AD-A111 107 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 22/3
DESIGN OF AN ORBITAL ELEMENT ESTIMATOR USING RELATIVE MOTION DA--ETC(U)
DEC 81 J F ANTHONY
UNCLASSIFIED AFIT/GA/AA/81D-1 NL

1 OF 2
AD
A111107

LEVEL II

AD A111107

DTIC FILE COPY

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

82 02 18 020

AFIT/GA/AA/81D-1

DESIGN OF AN ORBITAL ELEMENT ESTIMATOR USING RELATIVE MOTION DATA

THESIS

AFIT/GA/AA/81D-1 John F. Anthony
1 Lt USAF

AFIT/GA/AA/81D-1

DESIGN OF AN ORBITAL ELEMENT ESTIMATOR USING RELATIVE MOTION DATA

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

John F. Anthony
1Lt USAF
Graduate Astronautical Engineering
December 1981

Approved for public release, distribution unlimited

Acknowledgements

I would like to thank Dr. William Wiesel for the knowledge and encouragement he gave me while completing this thesis and attending his classes. Dr. Wiesel's immense knowledge of astrodynamics and orbit determination, both in the theoretical and operational sense, greatly helped me and motivated me in pursuing this project.

The role of a typist in completing a thesis is far more important than most students acknowledge. The skill and perserverence exhibited by my typist, Ms. Niki Maxwell, proved to be invaluable in completing this paper.

Finally, I thank my fellow astronautical engineering classmates for their help and encouragement as fellow students and more importantly as my friends.

John F. Anthony

Contents

		Page
Acknowledgements		ii
List of Figures		iv
List of Tables		v
Abstract		vi
I.	Introduction	1
II.	Least Squares Estimation Theory	4
	Probability Theory	4
	Dynamics	6
	Linear Least Squares Estimation	7
	Nonlinear Least Squares Estimation	9
III.	Development and Design	13
	Classical and Delaunay Orbital Elements	13
	Equations of Motion and Relative Motion	16
	Range and Range Rate Measurements	21
	Truth Model	24
	Relative Orbital Element Estimator	25
IV.	Testing and Results	30
V.	Conclusions and Recommendations	42
Bibliography		47
Appendix A:	Vector/Matrix Components of the Least Squares Estimation Equation	48
Appendix B:	Test Case Data	57
Appendix C:	Truth Model Flowchart and Computer Program	61
Appendix D:	Relative Orbital Element Estimator Flowchart and Computer Program	66
Appendix E:	Subroutines Used in Computer Programs	86
Vita		95

List of Figures

Figure		Page
1	Gaussian Error Function......................	4
2	Orientation Orbital Elements.................	13
3	Perifocal Coordinate System..................	14
4	Relative Position Computation................	17
5	Relative Velocity Computation................	17
6	PQW and IJK Coordinate Systems...............	18
7	Range Computation............................	22
8	Range Rate Computation.......................	23
9	Case I, Satellites at Epoch..................	34
10	Case II, III, Satellites at Epoch............	34

List of Tables

Table		Page
1	Case I Orbital Elements	33
2	Case I Variance Comparison	37
3	Case II Variance Comparison	37
4	Case III Variance Comparison	38
5	Case I Measurement Standard Deviation Comparison	38
6	Case II Measurement Standard Deviation Comparison	39
7	Case III Measurement Standard Deviation Comparison	39

Abstract

A relative orbital element estimator is designed using least squares estimation. Range and range rate measurements are taken and a vector of Delaunay elements, relative to an interceptor satellite, is estimated using the nonlinear least squares equation. The estimator incorporates a state vector coordinate transformation from relative position and velocity to relative orbital elements. Two modes of the estimator program are designed, batch and sequential, where the sequential mode uses Bayes estimation. Three test cases are analyzed and indicate satisfactory performance. Problem areas include estimator dependence upon a highly accurate initial estimate of the element vector to start the estimation process. The orbits are restricted to noncircular for both satellites and the orbits must be non coplanar. Range and range rate data for the estimator are provided using a truth model.

DESIGN OF AN ORBITAL ELEMENT
ESTIMATOR USING RELATIVE MOTION DATA

I. Introduction

Tracking an orbiting satellite from ground based stations is the primary method for determining a satellite's set of orbital elements. In a rendezvous and intercept situation the target orbital elements relative to the primary or intercept satellite are needed to compute the maneuvers necessary to accomplish the intercept. This data can be provided by ground-based tracking stations or can be determined using an estimation process on board the interceptor that uses relative motion data of the target from the interceptor.

Most relative motion studies involve linearization of the equations of motion about a circular orbit. Developing an orbital element estimator using relative motion data is based upon the dynamics or equations of motion relating the relative motion of two nearby satellites. The most commonly used equations for relative motion are the Hill's equations (Ref 5:111). These equations are used to develop relative motion relationships between two spacecraft that are in

neighboring near-circular orbits. This restriction to near circular orbits limits the use of Hill's equations for use in an orbital element estimation process using relative motion. Eades and Drewry (Ref 4) develop a set of relative motion equations but restrict the orbits to the same orbital period and restrict one satellite to a circular orbit. The estimation process used by NASA in the Apollo Guidance Computer Rendezvous Filter is designed to estimate both the position and velocity state vector of the primary and target satellite (Refs 9, 3). Measurements used are range, range rate, azimuth, and elevation angles. There is no restriction to the type of orbits, but the Rendezvous filter does not estimate a set of orbital elements. A satellite-to-satellite orbit determination system designed by NASA is the Tracking and Data Relay Satellite System (TDRSS) (Ref 6). This system uses least squares estimation processing radar measurements from a geostationary, circular orbit, station. This system is linked to a ground station that processes the tracking data and therefore is not autonomous. Also, the tracking satellite is restricted to a circular orbit.

Past studies of relative motion have dealt with the near circular orbit case. No method of orbital element determination from an in-orbit platform using relative motion for noncircular orbit scenarios has been developed. Development of a system to determine orbital elements from a spacecraft has its advantages in that it means longer visa-

bility and observation time from space than from the ground tracking stations. Determining the orbital elements is necessary to plan and accomplish a rendezvous and intercept of two spacecraft independent of ground tracking facilities. Ground tracking data could be supplemented with the data determined by the orbiting tracking systems for selected targets.

The orbital element estimator designed in this study uses least squares estimation theory to estimate a set of relative orbital elements of a target from a primary spacecraft. Incorporated in the design is a state vector coordinate transformation from relative motion data to relative orbital elements. Using this transformation in the least squares estimation process is a unique concept in the design of this orbital element estimator.

II. Least Squares Estimation Theory

Least squares estimation is based upon probability theory, specifically the principle of maximum likelihood, where the best estimate of a state is the value at which the probability of it being the true state is maximized.

Probability Theory

Equation (1) introduces the Gaussian probability density function.

$$P(e) \quad = \quad ((2\pi)^{-1/2}/\sigma) \exp\,(-(e)^2/2\sigma^2) \qquad (1)$$

The probability density function gives the probability that an error, denoted e, will lie between e and δe as $P(e)\delta e$. Integrated over the interval of all possible error this function is normalized to unity, as shown in Equation (2).

$$\int_{-\infty}^{\infty} P(e)de \quad = \quad (2\pi)^{-1/2}\,\sigma^{-1} \int_{-\infty}^{\infty} \exp\,(-e^2/2\sigma^2)de = 1 \qquad (2)$$

A gaussian error function is shown graphically in Figure 1 and supplemented by Equation (3).

$$P(X) \quad = \quad (2\pi)^{1/2}\sigma^{-1} \exp(-(X-X_o)^2/2\sigma^2) \qquad (3)$$

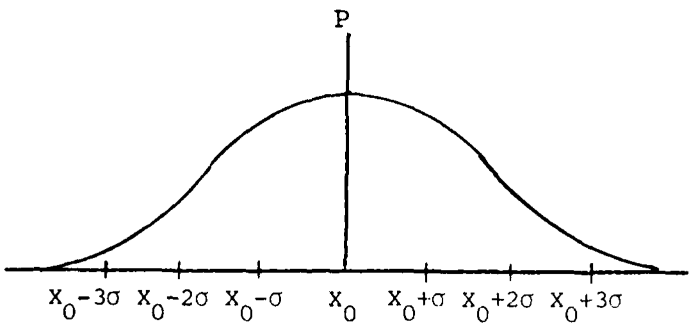


Fig 1. Gaussian Error Function.

The value X_o is the true measurement value. The standard deviation, σ, is a function of the accuracy of the measurement device. The Gaussian probability of an estimate being within 1σ of the true value is .68 or 68% probability.

The concept that links probability theory and least squares estimation is the principle of maximum likelihood. Simply stated, the best estimate of the true value is the estimated value which maximizes the probability of obtaining the true measurement.

The best estimate of X is $\hat{X}$. Substituting this for the true state yields the following probability function for N independent measurements, Equation (4), where $\bar{X}$ is the measurement vector and $\hat{\bar{X}}$ is the estimated measurement vector.

$$P(\bar{X}_i) = (2\pi)^{-N/2} \prod_i^N \sigma_i^{-1} \exp\left(-\sum_{i=0}^{N} (\bar{X}_i-\hat{\bar{X}}_i)^2/2\sigma_i^2\right) \quad (4)$$

To maximize the probability the absolute value of the argument of the exponential is minimized using Equation (5).

$$\frac{d}{d\hat{X}} \frac{\sum_{i=1}^{N}(\bar{X}_i-\hat{\bar{X}}_i)^2}{2\sigma_i^2} = \sum_{i=1}^{N} \frac{(\bar{X}_i-\hat{\bar{X}}_i)}{\sigma_i^2} = 0 \quad (5)$$

Minimizing $(\bar{X}-\hat{\bar{X}})^2$ is necessary to maximize the probability, hence the name least squares, where the sum of the squares of the errors is minimized. $(\bar{X}-\hat{\bar{X}})$ is called the residual, it is the difference between the observed measurement, X, and the estimate of X, again denoted $\hat{X}$.

Dynamics

Estimation of the state of a dynamical system, such as a set of satellite orbital elements, involves linearized dynamics. The estimate of the state, $\hat{X}$, is propagated in time by the vector differential shown in Equation (6).

$$\dot{\hat{\bar{X}}} = \bar{g}(\hat{\bar{X}}, t) \tag{6}$$

For trajectories of the state near the true trajectory, $\bar{X}=\bar{X}_o+\delta\bar{X}$, the dynamics of the variations in the trajectory is given in Equation (7),

$$\dot{\bar{X}} = \dot{\bar{X}}_o + \delta\dot{\bar{X}} = \bar{g}(\bar{X}_o + \delta\bar{X}, t) \tag{7}$$

Using a Taylor series expansion yields Equation (8).

$$\dot{\bar{X}}_o + \delta\dot{\bar{X}} = \bar{g}(\bar{X}_o, t) + \bar{\nabla}g(\bar{X}_o, t)\delta\bar{X} + O(2) \tag{8}$$

Equation (8) reduces to Equation (9) and introduces the A matrix which propagates the dynamics of the variations.

$$\delta\dot{\bar{X}} = A(t)\delta\bar{X},$$

where

$$A_{ij}(t) = \nabla g(\bar{X}_o, t) = \left.\frac{\partial g_i}{\partial X_j}\right|_{\bar{X}_o} \tag{9}$$

The state transition matrix, Φ, propagates the variations in the state as shown in Equation (10).

$$\delta X(t) = \Phi(t, t_o)\ \delta X(t_o) \tag{10}$$

The dynamics of the state transition matrix are determined by Equation (11).

$$\dot{\Phi}(t,t_o) = A(t)\Phi(t,t_o) \qquad (11)$$

The covariance, denoted P(t), is a matrix containing the average squared errors of the states related to themselves and other elements. The state transition matrix propagates the covariance by Equation (12).

$$P(t) = \Phi(t,t_o)\ P\ (t_o)\ \Phi^T\ (t,t_o) \qquad (12)$$

Linear estimation can be formulated using the linearized dynamics and subsequently related to nonlinear estimation, which is commonly used in astrodynamical problems.

Linear Least Squares Estimation

For a linear dynamical system the state transition matrix propagates the state from an epoch time, t_o, to a future time, t. The state at epoch is to be estimated from data measurements, $\overline{Z}_i$, taken over some time interval. The state vector is related to the measurement vector by Equation (13) where H_i is the observation relation matrix and e_i represents the error in the measurements.

$$\overline{Z}_i = H_i\ \overline{X}(t) + \overline{e}_i \qquad (13)$$

Associated with the measurement vector $\overline{Z}$ is the Q matrix which contains the squares of the σ values of the measurement device.

Equation (13) is rearranged to find an expression for the error between measurement and the estimated measurement, the result is Equation (14).

$$\overline{e}_i = \overline{z}_i - H_i\overline{X}(t_o) \tag{14}$$

The state vector X(t) is related to the state at epoch time using the state transition matrix propagation $\overline{X}(t_o)$ to $\overline{X}(t)$. Incorporating this into Equation (14) yields Equation (15).

$$\overline{e}_i = \overline{z}_i - \mathcal{H}_i\overline{X}(t_o) \tag{15}$$

where

$$\mathcal{H}_i = H\Phi(t_i, t_o)$$

Substituting Q_i as σ^2 and the error equation (Equation (15), into the Gaussian error function, Equation (4), yields an expression for the probability of error in the measurement set (Equation 16).

$$P(\overline{e}) = (2\pi)^{-N/2} \left| Q \right|^{-1/2} e^{(-1/2J)} \tag{16}$$

where $J = \overline{e}^T Q^{-1} \overline{e}$.

To maximize the probability, J is minimized in the same way the argument of the exponential was minimized in Equation (5).

$$\frac{\partial J}{\partial \hat{\bar{X}}} = \frac{\partial}{\partial \hat{\bar{X}}} \quad (\bar{Z} - \hat{\bar{X}}_o)^T Q^{-1} (\bar{Z} - \hat{\bar{X}}_o) \quad = 0 \qquad (17)$$

where $\hat{X}$ is the estimate of the state vector

Differentiating Equation (17) yields Equation (18).

$$\mathcal{H}^T Q^{-1} \mathcal{H} \hat{\bar{X}} - \mathcal{H}^T Q^{-1} \bar{Z} = 0 \qquad (18)$$

Rearranging Equation (18) leads to an expression which is the fundamental linear least squares equation.

$$\hat{\bar{X}}(t_o) = (\mathcal{H}^T Q^{-1} \mathcal{H})^{-1} \, {}^T Q^{-1} \bar{Z} \qquad (19)$$

where $(\mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$ is the covariance matrix.

<u>Nonlinear Least Squares Estimation</u>

Nonlinear estimation is based upon linear estimation theory using the linearized dynamics and modified components of linear estimation.

The measurements are now a nonlinear combination of the states and are given by the observation relation,

$$\bar{Z}(t_i) = \bar{G}(\bar{X}(t_i),(t_i)) \qquad (20)$$

This relation is linearized to give Equation (21).

$$\bar{e} = \frac{\partial \bar{G}}{\partial \bar{X}_i} \delta \bar{X}(t_i), \quad \frac{\partial \bar{G}_i}{\partial \bar{X}_i} = H_i \tag{21}$$

H_i is evaluated on the estimated or reference trajectory.

The residuals are given by Equation (22) using the nonlinear observation relation.

$$\bar{r}_i = \bar{Z}_i - \bar{G}(\bar{X}_{ref}, t_i) \tag{22}$$

Following the development of the linear least squares equation, using the nonlinear terms developed, gives the fundamental nonlinear least squares equation, Equation (23).

$$\delta \bar{X}(t_o) = (\mathcal{H}^T Q^{-1} \mathcal{H})^{-1} \mathcal{H}^T Q^{-1} \bar{r} \tag{23}$$

and the estimated state vector is given as Equation (24).

$$\bar{X}(t_o) = \bar{X}_{ref}(t_o) + \delta \bar{X}(t_o) \tag{24}$$

Nonlinear least squares Equations (23) and (24) are used to update the reference state vector iteratively until it converges upon a solution. The convergence criteria is based upon the true solution being the result. Theoretically $\delta \bar{X}(t_o)$ will converge to zero, in practice it should be allowed to converge to well within the associated square root of the covariance for that element, $\sqrt{P_{ii}}$. The residuals, independently, should be of order $\sqrt{Q_{ii}}$ as they converge.

The algorithm below shows the step by step iterative process used to converge upon a solution.

NONLINEAR LEAST SQUARES ALGORITHM

1. From each measurement calculate:

   a. $\overline{r}_i = \overline{Z}_i - G(\overline{X}_{ref}(t_i), t_i)$

   b. H_i

   c. $\Phi\ (t_i, t_o)$

2. Assemble vector/matrices necessary for nonlinear least squares equation.

$$\overline{r} = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_i \end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix} H_1\ \Phi_1 \\ H_2\ \Phi_2 \\ \cdot \\ \cdot \\ \cdot \\ H_i\ \Phi_i \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 & & & & \\ & Q_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & Q_i \end{bmatrix}$$

3. Compute update using least squares equation

$$\delta\hat{\overline{X}}(t_o) = (\mathcal{H}^T Q^{-1} \mathcal{H})^{-1}\ {}^T Q^{-1} \overline{r}$$

4. Update reference solution.

$$\hat{X}(t_o) = \overline{X}(t_o) + \delta\hat{\overline{X}}(t_o)$$

5. Convergence Check! If convergence criteria is met $\hat{X}(t_o)$ is the solution. If not met, return to Step 1 where, using the observation relation, a new set of residuals are computed.

Nonlinear least squares estimation is implemented with a batch of measurements using the given algorithm. Combining sequential measurements with previous estimates of

a state is also accomplished using the fundamental equation.

A sequential nonlinear least squares process is implemented using the previous estimate of the state, denoted $\bar{X}^-$, as data with an associated covariance, denoted $P^-(t_o)$. Since the estimate is at epoch time the observation relation matrix for this data is the identity matrix. The components of the fundamental equation for the sequential least squares are given in Equation (25).

$$\mathcal{H} = \begin{bmatrix} I \\ \hline \mathcal{H}_{seq} \end{bmatrix} , \quad Q = \left[\begin{array}{c|c} P^- & \\ \hline & Q_{seq} \end{array}\right] , \quad (25)$$

$$\bar{r} = \begin{bmatrix} \hat{\bar{X}} - \bar{X}_{EST} \\ \hline \bar{r}_{seq} \end{bmatrix}$$

Substituting these into the fundamental equation for nonlinear estimation gives Equation (26).

$$\delta\hat{\bar{X}} = \left(P^-(t_o) + \mathcal{H}^T Q^{-1} \mathcal{H}\right)^{-1} \left(P^-(t_o)(\hat{\bar{X}}^- - \bar{X}) + \mathcal{H}^T Q^{-1} \bar{r}\right) \quad (26)$$

where $\hat{\bar{X}}^-$ is the previous state estimate.

Sequential least squares is more commonly called Bayes estimation. Bayes estimation minimizes the squares of the sequential residuals and the discrepancy between the previous estimate and the estimate of the state vector determined by the Bayes estimation process. The Bayes estimation equation is used iteratively which is similar to the given least squares algorithm.

III. Design and Development

Classical and Delaunay Orbital Elements

The orientation and size of a satellite orbit is described by a set of orbital elements. The classical orbital elements are the most commonly used. This estimator uses the Delaunay orbital elements which can be directly related to the classical elements. The Delaunay elements are the standard set of canonical elements for the two body problem. All element sets are mathematically equivalent, and therefore one set can be related to another set.

To best describe the Delaunay elements the meaning of the classical elements is necessary. Figure 2 gives the classical orientation elements.

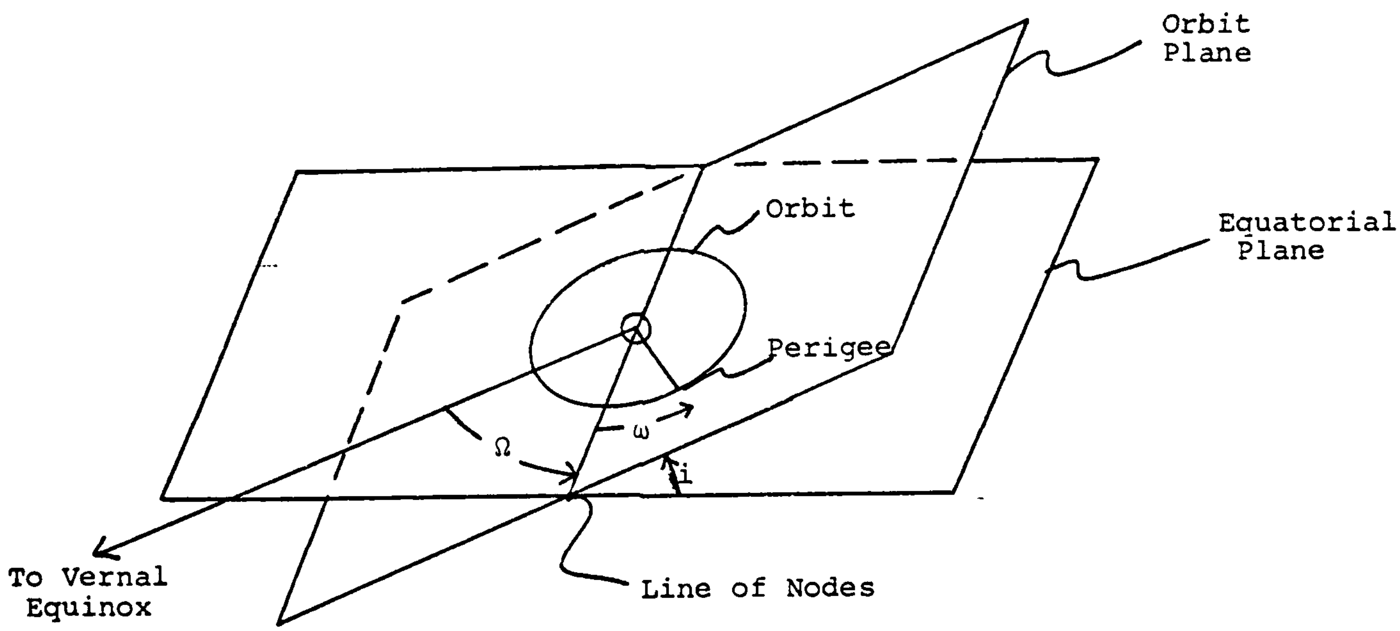


Fig 2. Orientation Orbital Elements.

The longitude of ascending node is Ω, i is the inclination to the equator plane, and ω is the argument of perigee. The vernal equinox is the direction in space used as a reference for the inertial coordinate system used in this design. The dimensional elements specify the size and shape of the orbit and relate position in the orbit with time. The semi-major axis is denoted a, e is the eccentricity, it defines the shape of the orbit and M is the mean anomaly which determines the satellite's position in the orbit. Mean anomaly is an angular element. Commonly used with the classical elements is a coordinate system that has unit vector $\hat{P}$ directed to perigee, the closest point in the orbit to the attracting body. $\hat{Q}$ is perpendicular to $\hat{P}$ and $\hat{W}$ completes the orthogonal right handed set. This frame is called the perifocal coordinate system, or PQW, as shown in Figure 3.

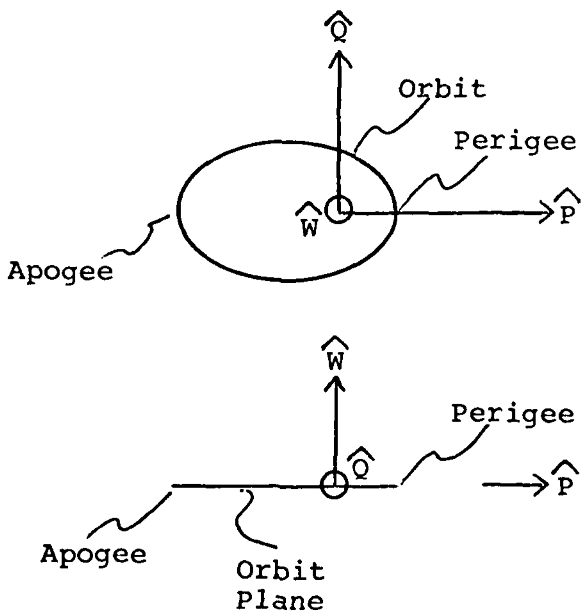


Fig 3. Perifocal Coordinate System.

The Delaunay elements can be related to the classical elements using Equations (1) through (6) (Ref 11:24).

$$L = \sqrt{\mu a}, \quad \text{where } \mu \text{ is the gravitational constant} \tag{1}$$

$$l = n(t-T_{PERIGEE}) = M, \text{ where } n = \sqrt{\frac{\mu}{a^3}} \tag{2}$$

$$G = \sqrt{\mu a(1-e^2)} = L\sqrt{1-e^2} \tag{3}$$

$$g = \omega \tag{4}$$

$$H = \sqrt{\mu a(1-e^2)} \cos i = G \cos i \tag{5}$$

$$h = \Omega \tag{6}$$

Delaunay elements g and h are identical to the classical orientation elements ω and Ω. Element l is the mean anomaly and is also defined by Equation (7) which introduces E, the eccentric anomaly, an angular measurement from the center of the ellipse.

$$l = E - e \sin E \tag{7}$$

Element L is related to the semi-major axis and used to define the total energy of the orbit. G is related to the eccentricity and is the total angular momentum of the orbit. H is the inertial vertical component of the angular momentum.

The classical orbital elements in terms of the Delaunay elements are given in Equations (8) through (13).

$$a = L^2/u \tag{8}$$

$$e = \sqrt{1 - G^2/L^2} \tag{9}$$

$$i = \cos^{-1}(H/G),\ \sin^{-1}(1-H^2/G^2) \tag{10}$$

$$\omega = g \tag{11}$$

$$\Omega = h \tag{12}$$

$$M = l \tag{13}$$

Equations of Motion and Relative Motion

Relative motion can be described as the apparent motion of another object as seen from the primary object. The relative motion of two nearby satellites is found by determining the position and velocity of each vehicle and differencing the vectors accordingly to find the relative position and velocity. The position and velocity must be in a common coordinate system. The position and velocity of a satellite is a function of the orbital elements and time. Figures 4 and 5 show how the relative position and velocity vectors are found. (Figures 4 and 5 follow on the next page.)

Position and velocity computations for each satellite are initially computed in the PQW coordinate system. To find the inertial relative motion the vectors are transformed into an inertial frame, defined IJK, where I is directed to the vernal equinox. Figure 6 shows the relationship between PQW and IJK coordinate frames.

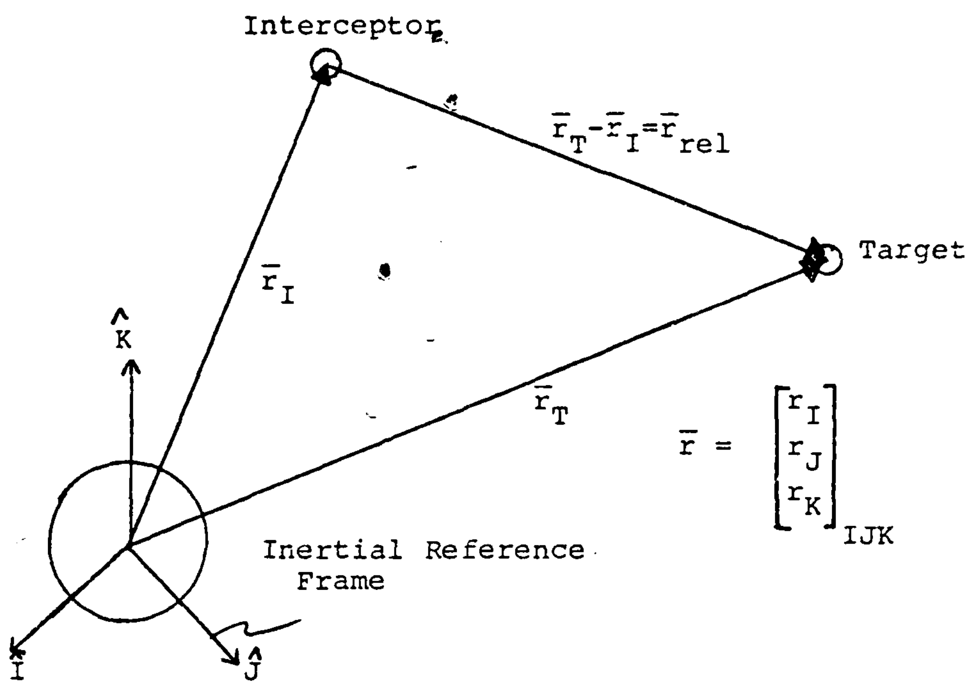


Fig 4. Relative Position Computation.

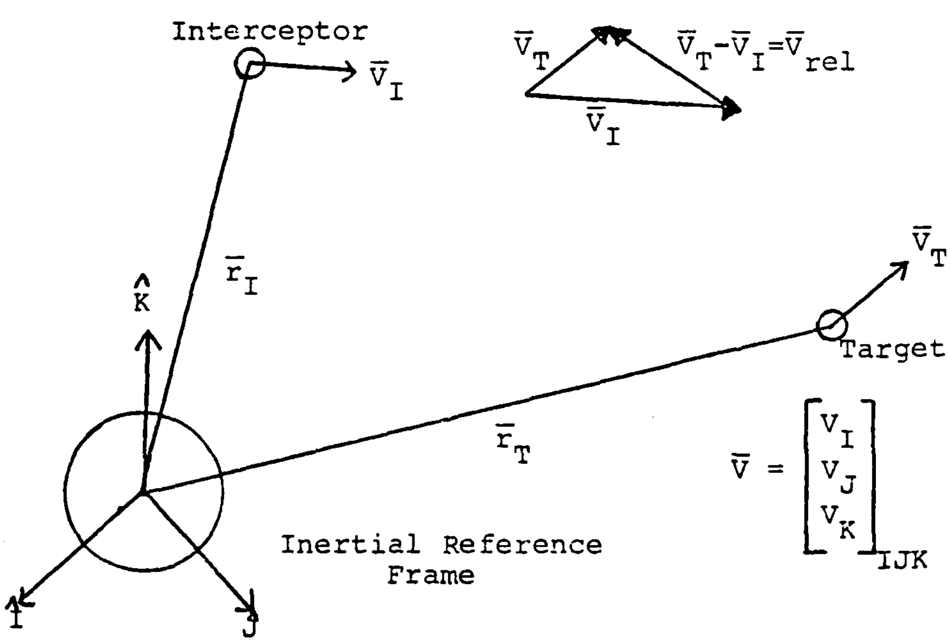


Fig 5. Relative Velocity Computation.

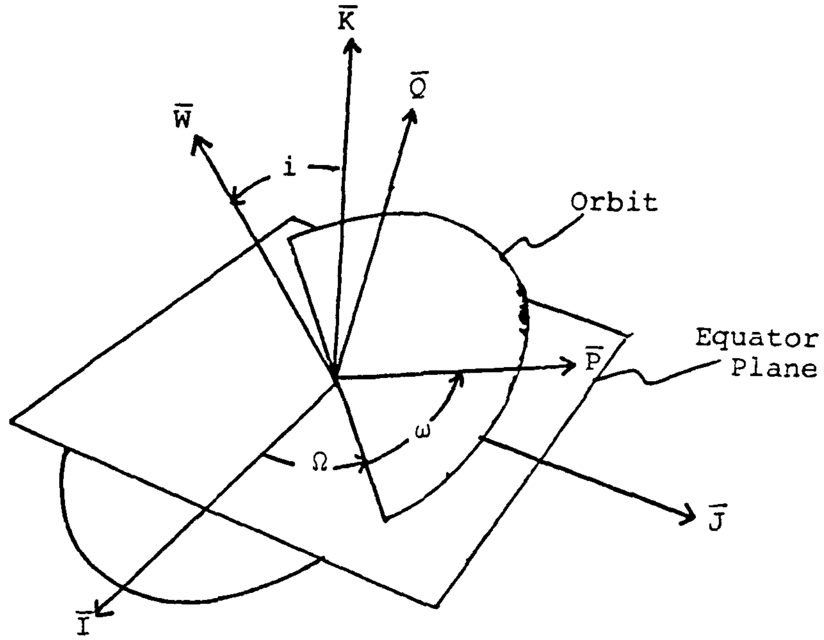


Fig 6. PQW and IJK Coordinate Systems (Ref 1:80).

Position in the PQW system (Ref. 1:80) is given by Equations (1) and (2).

$$r_P = (L^2/\mu)(\cos(E) - \sqrt{1-G^2/L^2}) \quad (1)$$

$$r_Q = (LG/\mu)\ \sin(E) \quad (2)$$

where E is the eccentric anomaly and the initial E is assumed to be zero.

Velocity, in the PQW system, is found by taking the time derivative of the position vector, shown in Equation (3).

$$\bar{v}_{PQW} = \frac{d}{dt}\ \bar{r}_{PQW} = \frac{d\bar{r}_{PQW}}{dE} \cdot \frac{dE}{dl} \cdot \frac{dl}{dt} \quad (3)$$

The mean motion is defined as dl/dt and is given in Equation (4).

$$\frac{dl}{dt} = \mu^2/L^3 \tag{4}$$

dE/dl is the change in the eccentric anomaly with respect to l, the mean anomaly. Differentiating implicitly yields Equation (5).

$$dl = dE - e\cos(E)dE \tag{5}$$

Substituting e in terms of the Delaunay elements in Equation (6),

$$dl = dE - \sqrt{1-G^2/L^2}\cos E\, dE \tag{6}$$

and separating leads to Eq (7)

$$\frac{dE}{dl} = \frac{1}{1 - \sqrt{1-G^2/L^2}\cos E} \tag{7}$$

$d\bar{r}/dE$ is found by differentiating the vector components of position with respect to the eccentric anomaly E. Equations (8) and (9) are the results.

$$dr_P/dE = (-L^2/\mu)\sin(E) \tag{8}$$

$$dr_Q/dE = (LG/\mu)\cos(E) \tag{9}$$

Substituting expressions from Equations (4), (7), (8), and (9) into (3) gives expressions for the velocity vector in the PQW frame, Equations (10) and (11).

$$V_P = -\mu \sin(E)/(L(1 - \sqrt{1-G^2/L^2} \cos(E))) \quad (10)$$

$$V_Q = \mu G \cos(E)/(L^2(1 - \sqrt{1-G^2/L^2} \cos(E))) \quad (11)$$

The position and velocity state vector is given as a six-component vector Equation (12).

$$\overline{X}_{PQW} = \begin{bmatrix} r_P \\ r_Q \\ r_W \\ V_P \\ V_Q \\ V_W \end{bmatrix} , \quad \text{where } r_W \text{ and } V_W = 0$$

Rotation from PQW to IJK frame is accomplished using Equation (13), where R, the transformation matrix from PQW to IJK is given in Appendix A.

$$\overline{X}_{IJK} = [R]\overline{X}_{PQW} \quad (13)$$

The position and velocity of each satellite is computed at given times and differentiated accordingly in the inertial frame to yield the relative position and velocity

vector of the target from the interceptor (Equation 14).

$$\overline{X}_{RELATIVE} = \begin{bmatrix} r_{T,I} - r_{I,I} \\ r_{T,J} - r_{I,J} \\ r_{T,K} - r_{I,K} \\ V_{T,I} - V_{I,I} \\ V_{T,J} - V_{I,J} \\ V_{T,K} - V_{I,K} \end{bmatrix} = \begin{bmatrix} \Delta r_I \\ \Delta r_J \\ \Delta r_K \\ \Delta V_I \\ \Delta V_J \\ \Delta V_K \end{bmatrix} \qquad (14)$$

T = Target

I = Interceptor

Range and Range Rate Measurements

Range and range rate measurements are used as observational data for the estimator. These measurements were selected since they relate directly to relative position and velocity and are typically measured by a radar unit. The radar unit used in the design has a 100 meter standard deviation, σ, in range accuracy and a 0.3 meter per second σ in range rate accuracy. A radar's accuracy is dependent upon several factors. Range to the target and power provided to the radar are basically those factors that affect the accuracy and capability. Cross section of the target and shape are also important factors to consider. This estimator uses a theoretical radar that is not range restricted and has the constant σ values given previously. It is assumed that the interceptor radar is able to track any target, regardless of size, shape, or range. Special cases tested used a range restriction of less than 200 Km,

which is a reasonable range limit for a radar. Generally, the range was not restricted for testing the performance and accuracy of the estimator.

Range is measured along a line of sight vector from the interceptor to the target. Range is the magnitude of this line of sight vector, as shown in Figure 7.

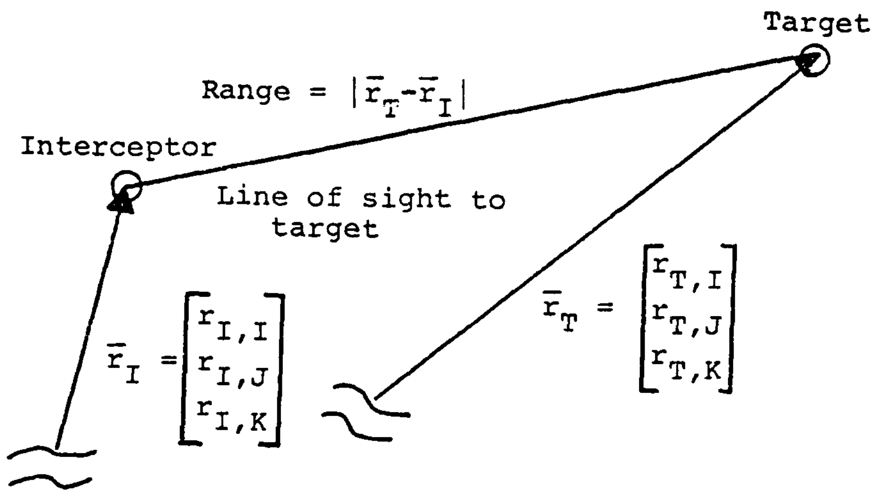
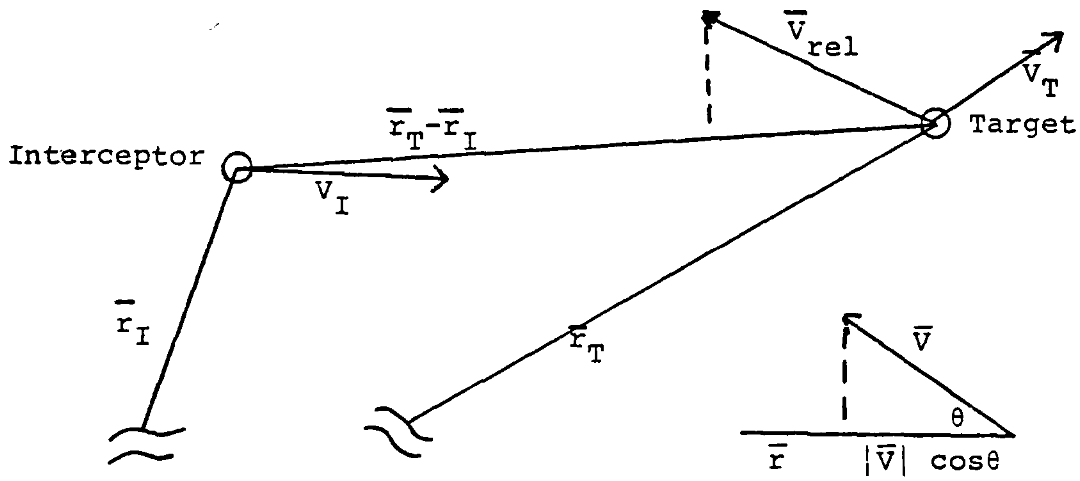


Fig 7. Range Computation.

In terms of the relative position the range is given in Equation (1).

$$\text{Range} = (\Delta r_I^2 + \Delta r_J^2 + \Delta r_K^2)^{1/2} \qquad (1)$$

Range rate is the rate of change of the range scalar. The dot product of the relative velocity vector and the relative position vector divided by range gives range rate, as shown in Figure 8.



Range Rate = $(\bar{V} \cdot \bar{r})/|\bar{r}| = |\bar{V}| \cos\theta$

Fig 8. Range Rate Computation.

Equation (2) gives the range rate in terms of the relative position and velocity.

$$\text{Range rate} = (\Delta V_I \Delta r_I + \Delta V_J \Delta r_J + \Delta V_K \Delta r_K)/\text{Range} \quad (2)$$

The estimator program reads the range in distance units (DU), where 1 DU = 6378.145 Km, range rate in distance units per time unit (DU/TU), where 1 DU/TU = 7.905376 Km/sec, and time in TU, where 1 TU = 806.8136 seconds. The time used is

the time past epoch, the epoch time being when the interceptor was a perigee. (Ref. 1:429)

Truth Model

The truth model generates the true range and range rate data at given times for various orbital scenarios. The truth model data is generated using a computer program that propagates the position and velocity of two satellites, an interceptor and target, in two orbits. The position and velocity are propagated using the equations previously developed, Equations (1), (2), (10), and (11) of the Equations of Motion section of this chapter. Time is the common variable that is synchronized for both satellites. Each satellite is started from an initial position and moved using 1 minute time increments. At each time the associated eccentric anomaly is computed for the equation of motion. Orbit size and orientation are computed using a given set of orbital elements for each satellite. The inertial position and velocity vector at each measurement time are computed and differenced accordingly to give the relative position and velocity. Range and range rate are computed using the relative position and velocity and are given by Equations (1) and (2), which are repeated from the Range and Range Rate section.

$$\text{Range} = (\Delta r_I{}^2 + \Delta r_J{}^2 + \Delta r_K{}^2)^{1/2} \qquad (1)$$

$$\text{Range Rate} = (\Delta V_I \Delta r_I + \Delta V_J \Delta r_J + \Delta V_K \Delta r_K)/\text{Range} \qquad (2)$$

The truth model computer program gives range, range rate, and time at 10 minute intervals for an orbit scenario, but minor modifications can adjust the time interval. The computer program and associated flow chart for the truth model are presented in Appendix C.

Relative Orbital Element Estimator

The estimator designed in this study uses least squares estimation theory to estimate a set of relative orbital elements. The relative orbital element state, Equation 1, is the difference between the known interceptor elements and the estimated target elements at an epoch time, here chosen to be the time when the interceptor is at perigee prior to taking range and range rate measurements. Initially an estimate of the relative element vector is needed to start the estimation process. In this study the initial estimate is found by perturbing the true relative element vector by a certain percentage selected by the user.

$$\overline{X}_{ELEMENTS} = \begin{bmatrix} \delta L \\ \delta l \\ \delta G \\ \delta g \\ \delta H \\ \delta h \end{bmatrix} \qquad (1)$$

The estimator is designed using the developed theory, however, a modification unique to the design is incorporated. The modification used in this estimator that is different from typical least squares estimation is the state vector coordinate transformation from relative position and velocity to a relative orbital element state vector. Variations in the relative position and velocity are related to the variations in the Delaunay orbital elements. This modification is incorporated via the state transition matrix.

From the Least Squares Theory section, Equation (2) depicts the classical way the Φ matrix is used to propagate the variations of the state vector in time.

$$\delta\overline{X}(t) = \Phi(t,t_o)\delta\overline{X}(t_o) \tag{2}$$

The Φ used in this design is not classical. Here Φ propagates the variations of the position and velocity relative to the interceptor orbit. These variations are propagated from an epoch time to a particular time, in this case a measurement time. As well as propagating motion, Φ also relates the variations in position and velocity to variations in the Delaunay elements. Thus, Φ, called the state transition/state vector coordinate transformation matrix, relates relative position and velocity to the relative orbital elements. Φ is formed by developing relationships between the satellite position and velocity in the

PQW coordinate frame and the associated Delaunay orbital elements and then transforming the result into the IJK coordinate frame. The position and velocity vector (Equation 3) and the orbital elements vector (Equation 4) are related to each other in the PQW reference frame by the relation given in Equation (5)

$$\bar{X}_{r,V} = [r_P,\ r_Q,\ r_W,\ V_P,\ V_Q,\ V_W] \tag{3}$$

$$\bar{X}_{ELEMENTS} = [L,\ l,\ G,\ g,\ H,\ h] \tag{4}$$

$$\left.\frac{d(\bar{X}_{rv})}{d(\bar{X}_{ELM})}\right|_{PQW} = \frac{d(\bar{X}_{r,v})}{dE} \cdot \frac{dE}{d(\bar{X}_{ELM})} + \frac{\partial \bar{X}_{r,v}}{\partial \bar{X}_{ELM}} \tag{5}$$

Equation (5) results in a 6x6 matrix relating variations of the six components of position and velocity to variations of the six orbital elements in the PQW frame. The result of Equation (5) is transformed into the IJK coordinate frame using Equation (6).

$$\frac{d(\bar{X}_{r,V})_{IJK}}{d(\bar{X}_{ELM})} = \frac{\partial R}{\partial(\bar{X}_{ELM})}\ \bar{X}_{r,V} + [R]\ \frac{d(\bar{X}_{r,V})}{d(\bar{X}_{ELEM})} \tag{6}$$

R is a coordinate rotation matrix from the PQW to IJK frame. The result of Equation (6) is a 6x6 matrix relating the variations of relative position and velocity to variations in the Delaunay elements in the IJK frame. The detailed vector and matrix components for the development of Φ as well as the other components necessary for implemen-

tation of the nonlinear least squares equation are presented in Appendix A.

The validity and accuracy of Φ is checked using numerical derivatives and comparing them with the appropriate column vector of Φ at the given time. A position and velocity vector is determined using a given set of orbital elements. One element is perturbed a small amount, δ, and another position and velocity vector is found. The vector difference divided by the change of the element should agree with the associated column vector of Φ. Equation (7) depicts the check method.

$$\frac{\bar{X}_{\delta} - \bar{X}}{\delta_{\text{ELEMENT}}} = \frac{d\bar{X}}{d\delta_{\text{ELEMENT}}} \tag{7}$$

After Φ was formulated several checks using various orbit types indicated the validity and accuracy of Φ.

Incorporating the state vector coordinate transformation into this estimator meant that relative position and velocity data, easily measured, could be directly transformed to the orbital elements necessary for orbital determination. The estimator computer program and flowchart are presented in Appendix D.

The estimator is checked using data generated by the truth model. Various scenarios are simulated where range and range rate measurements of a target satellite are made from an interceptor satellite at selected time intervals. A batch mode and a sequential mode are available where an

estimation of the relative orbital elements is made based on a single batch of measurements or several sequential batches taken in time segments. The sequential mode uses Bayes estimation theory. Flexibility in the truth model, its ability to generate data for any interceptor-target scenario, leads to flexibility in the ways the estimator can be tested, therefore revealing strong points or weaknesses in the accuracy and versatility of the design. The Testing and Results section of this report presents the statistics and discusses the various scenarios and special tests performed using the relative orbital element estimator program.

IV. Testing and Results

The performance of the estimator in the test cases is evaluated by comparing the statistics of the estimated element vector to the statistics predicted by the estimator.

The covariance matrix $(\mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$ for the batch least squares and $(P(-)^{-1} + \mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$ for the Bayes or Sequential estimation give the estimators prediction of how well it can estimate the state vector or relative orbital elements. The diagonal elements of the covariance matrix give the variances, σ^2, for each element. These variances are indicators of the estimator's predicted performance. From the estimated relative orbital elements resulting from a set of simulations, where random noise corrupted the measurements, a sample variance is computed using Equation (1), which is the definition of a sample variance.

$$\sigma_e^2 = \frac{1}{N-1} \sum_{J=1}^{N} (\bar{X}_{J,e\ \mathrm{ESTIMATED}} - \bar{X}_{e,\mathrm{TRUE}})^2 \qquad (1)$$

where

$\bar{X}$ is the relative element state vector

N is the number of simulations

e denotes the Delaunay Element

The variances computed using Equation (1) are the results of the estimations made in the simulations for a particular scenario. Similarity between these actual element variances and the variances predicted by the estimator from the covariance matrix indicate satisfactory performance

of the estimator.

The measurement device accuracy over all ranges to the target is given in Equations (2) and (3).

$$\sigma_{RANGE} = 1.57X10^{-5} \text{ DU} = 100 \text{ meters} \quad (2)$$

$$\sigma_{RANGE\ RATE} = 3.85X10^{-5} \text{ DU/TU} = 0.3 \text{ meters/second} \quad (3)$$

These values indicate the accuracy limits to which the measurement device operates. From the residuals an indication of the accuracy to which the estimator updated the predicted measurement can be found and compared to the given radar accuracies.

The root mean squared value (RMS) of the residuals are computed in each simulation using Equation (4), the definition for RMS value for a sample group of measurements.

$$\text{Measurement RMS} = \frac{1}{N} \sqrt{\Sigma\ (\bar{r}_m - \bar{r}_p)^2} \quad (4)$$

where

r_m is the measurement

r_p is the estimator predicted measurement

N is the number of measurements taken.

The variances for the range and range rate residuals are computed using Equation (5) using the measurement RMS values from Equation 4.

$$\sigma^2 = \frac{1}{N_T - 1} \sum_{k=1}^{N_R} \left(\sum_{i=1}^{N_O} (RMS_i)^2 \right)_k \quad (5)$$

where

$N_T = N_R N_O$

N_R is the number of simulations

N_O is the number of observations per simulation

The square root of the variance gives the standard deviation. The standard deviation of the residuals, computed using Equation (5), are compared to the given standard deviation of the radar accuracies. Agreement of the associated standard deviations indicates the estimator has predicted the range and range rate measurements to the accuracy of the measurement device.

Three test case scenarios are statistically evaluated to check the performance of the estimator. Simulations of each test case are made using random noise inputs to the measurement data. The initial estimate used to start the estimation process is the true relative element vector for that particular test. The true vector is used because difficulties are experienced when a perturbed initial element vector is used to start the estimator. These difficulties in the requirement of a highly accurate initial estimate are discussed shortly in this section. The test cases are now described.

Case I uses the least squares batch mode of the estimator. In this mode a number of range and range rate measurements are taken and processed to give estimate of the relative element vector. Case I uses 13 observations over a

1 orbit period of the interceptor. Table 1 gives Case I classical orbital elements and relative Delaunay elements.

TABLE 1

CASE I ORBITAL ELEMENTS

	Interceptor	Target
a	1.3 DU	1.35
e	0.2	0.3
i	45°	46°
Ω	45°	46°
w	45°	46°
l_o	0°	0°

ΔL	0.2171958	DU^2/TU	Δl	0	RAD
ΔG	-0.00876201	DU^2/TU	Δg	.0174533	RAD
ΔH	-.0199932	DU^2/TU	Δh	.0174533	RAD

l_o is the mean anomaly at the epoch time. The epoch time is selected as the time at which the relative orbital elements are to be estimated. Figure 9 shows the orbit configuration at epoch time for Case I. Case II also uses the least squares batch mode. The primary difference is that in Case II at epoch time there is a 0.1 radian difference in the mean anomaly. In Case I at epoch the two satellites are co-linear on the P vector of the PQW coordinate frame while in Case II there is an out of phase or non co-linear situation at epoch as shown in Figure 10. Case II elements are identical to Case I except the target l_o is 0.1 radian so the Δl at each epoch is 0.1 radian.

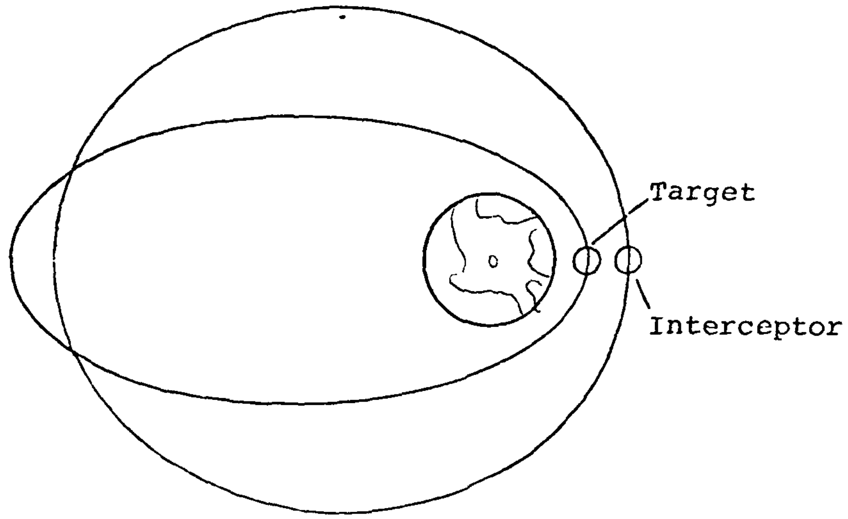


Fig 9. Case I, Satellites at Epoch.

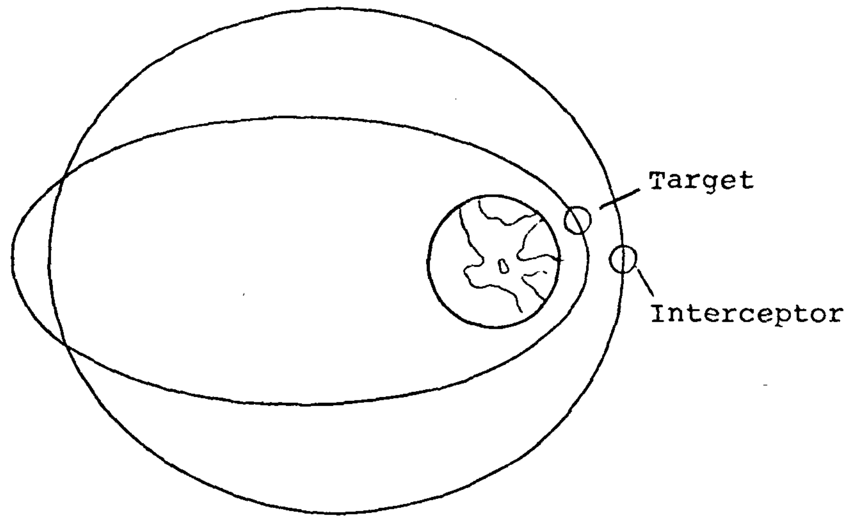


Fig 10. Case II, III, Satellites at Epoch.

Case III uses the sequential mode of the estimator. In this mode a least squares batch estimation based on an initial set of observations is made. This is an initial set of 13 observations. This is followed by sequential sets of observations where two sets of three measurements over a 30 minute time segment are made. After the initial set and two sequential sets of measurements are taken, the relative orbital elements are estimated. Case III elements are identical to Case II.

Appendix B presents the measurement data for each test case. This data is obtained from the truth model, in the simulations random noise corrupts the measurements. An explanation of the random noise inputs to the data is also contained in Appendix B.

Statistics for Case I are based on 25 simulations, Case II is based on 30, and Case III is calculated from 18 simulations. In each simulation different sequences of random noise inputs are used. As noted earlier in this section, the estimator uses the true relative element vector in all evaluated test cases as an initial estimate to start the estimation process, therefore, the test cases presented indicate the estimator error in estimating a solution for the relative element state vector, when given the true state vector. The statistics for the test cases are included in Tables 2 through 7, shown on the following pages. Contained in the first column of Tables 2, 3, and 4 are the variances

computed from Equation (1) using the resultant vectors from the associated case simulations. The second column contains the average variances for each element from the covariance matrix for that case. Similarly in the first column of Tables 5, 6, and 7 is the standard deviation computed using Equation (5) and the second column contains the given parameters for the radar.

The comparison of statistics indicates good agreement between the actual and predicted statistics. The estimator predicted statistics agree with the statistics of the residuals and estimated element vector. This agreement tells us that the estimator is functioning optimally for the given scenarios and initial conditions. Comparing magnitudes of the variances show that σ^2_{LL} and σ^2_{GG} are smaller than the other element variances. This indicates the estimator's ability to determine the dimensional element L and G with more confidence or accuracy than the angular elements l, g, h, H. In comparing Case II and III variances, where the conditions at epoch are the same, but Case II uses the batch mode while Case III incorporates more data using the sequential mode, the variances for Case III are notably smaller than Case II. This decrease in the covariance from the batch to sequential case indicates the advantage of incorporating more data sequentially. A decrease in the covariance implies more confidence in the estimator's pre-

TABLE 2

CASE I VARIANCE COMPARISON

Element	σ_{ii}^2 From Element Vector Equation (1)	σ_{ii}^2 From Covariance Matrix
L	2.4 E-12	3.0 E-12
l	2.6 E-9	9.8 E-10
G	1.35 E-11	7.0 E-12
g	2.11 E-8	3.9 E-8
H	3.55 E-8	1.4 E-8
h	1.6 E-8	6.2 E-8

Units L, G, H = DU^2/TU l, g, h = radians

TABLE 3

CASE II VARIANCE COMPARISON

Element	From Element Vector σ_{ii}^2 Equation (1)	σ_{ii}^2 From Covariance Matrix
L	4.02 E-12	1.7 E-11
l	2.17 E-9	7.6 E-10
G	2.09 E-10	3.5 E-12
g	2.46 E-8	2.9 E-8
H	2.02 E-8	2.0 E-8
h	4.22 E-8	5.0 E-8

TABLE 4

CASE III VARIANCE COMPARISON

Element	From Element Vector σ_{ii}^2 Equation (1)	From Covariance Matrix σ_{ii}^2
L	1.19 E-12	3.2 E-13
l	1.09 E-10	1.5 E-10
G	4.5 E-12	2.3 E-12
g	1.44 E-8	4.05 E-9
H	9.58 E-9	1.6 E-9
h	2.69 E-9	7.7 E-9

TABLE 5

CASE I MEASUREMENT STANDARD DEVIATION COMPARISON

Measurement	$\sigma_{R,RR}$ Computed from Residuals Equation (5)	$\sigma_{R,RR}$ Radar Parameters
Range	1.96 E-5 DU	1.57 E-5
Range Rate	3.96 E-5 DU/TU	3.85 E-5

TABLE 6

CASE II MEASUREMENT STANDARD DEVIATION COMPARISON

Measurements	$\sigma_{R,RR}$ Computed from Residuals Equation (5)	$\sigma_{R,RR}$ Radar Parameters
Range	1.74 E-5	1.57 E-5
Range Rate	3.77 E-5	3.85 E-5

TABLE 7

CASE III MEASUREMENT STANDARD DEVIATION COMPARISON

Measurement	$\sigma_{R,RR}$ Computed from Residuals Equation (5)	$\sigma_{R,RR}$ Radar Parameters
Range	1.55 E-5	1.57 E-5
Range Rate	3.53 E-5	3.85 E-5

diction of the element set when the Bayes estimator process is implemented.

Simulations using initial estimates other than the true vector were also tested. These initial estimates were computed using a perturbing factor that perturbed the true vector by a percentage determined by the magnitude of the perturbing factor. The sign of the perturbation, + or -, is determined by a random process. These simulations were made using noise free measurements and the initial estimate of the relative element vector was perturbed. The percentage of perturbation to the true element vector was limited to values less than or equal to 1%. When values greater than 1% were used the estimator would not converge to a solution. When noise corrupted measurements were used the limit of the perturbing percentage decreased to approximately 1/10%. This indicated that with noise corrupted measurements the estimator is unable to converge to a solution unless a highly accurate initial estimate of the element vector is provided.

Several orbit scenario cases indicated weaknesses in the estimator performance. If the target and interceptor orbits are coplanar the residuals will always diverge and convergence to a solution is impossible. The relative inclination in this case is zero, therefore the relative longitude of ascending node, h, is undefined. The equation for satellite dynamics and reference frame transformations contain h and in the coplanar case with h undefined a singu-

larity exists. This singularity occurs for low relative inclination, near zero, situations. For circular orbits perigee is undefined and in near circular orbits the perigee may be difficult to locate due to small changes in the distance from the attracting body throughout the orbit trajectory. Since the estimator uses a perigee dependent reference coordinate system the circular orbit weakness was recognized. Since the classical and Delaunay elements are dependent upon the existence of a perigee the estimator has difficulty estimating the angular elements l and g that need a defined perigee for definition. Simulations revealed that the estimator could not estimate l and g separately. This was indicated by large variances for the angular elements; however, the sum of l and g gave an accurate estimation of target angular position at epoch time. Therefore, in near circular cases determining angular position in the orbit is possible but determining the angular elements l and g separately results in erroneous values for those individual elements. The estimator estimates Delaunay elements L and G more accurately than the angular elements l, g, h, and H as indicated by the estimator predicted variances found in Tables 2 through 7.

V. Conclusions and Recommendations

Least squares estimation theory and techniques were used in the design of the relative orbital element estimator. The new and unique feature incorporated in this estimator is the state vector coordinate transformation from relative position and velocity coordinates to relative orbital elements. It is possible that the difficulties encountered in the orbit scenarios discussed in the results can be solved using several modifications or changes to the development and implementation of the design. Several solutions or recommendations are now presented.

Using the conventional set of Delaunay elements lead to some of the difficulties experienced with circular, near circular, and coplanar orbits of the interceptor and target. In a circular orbit perigee is undefined. With perigee undefined, the argument of perigee, g, is therefore undefined. Since the equations of motion and coordinate transformation from the perifocal to inertial coordinate system are dependent upon g as a variable these equations become incalculable. In near circular orbits the element g is difficult to define since the orbit is so near being circular and the problems of singularities in the circular orbit case arise. Coplanar orbits have the relative inclination equal to zero. In this situation, longitude of ascending node, h, is undefined. Again equations for motion

and transformation will contain an undefined term and be incalculable.

These problems can be solved using another set of orbital elements that can successfully handle low inclination, circular or near circular orbits. The ideal set for this is the Equinoctal orbital elements. These elements are used by the NORAD Space Computational Center (Ref 2) and are free from both zero inclination and eccentricity difficulties. The Equinoctal elements are given in Equations (1) through (6).

(1) $a_f = e \cos \pi$ where $\pi = \Omega + \omega$ is the longitude

(2) $a_g = e \sin \pi$ of perigee

(3) M = mean motion

(4) $L = \Omega + \omega + M = \pi + M$ = mean longitude when M is the mean anomaly

(5) $\chi = \dfrac{\sin i \sin \Omega}{1 + \cos i}$

(6) $\psi = \dfrac{\sin i \cos \Omega}{1 + \cos i}$

With circular orbit elements a_f and a_g are zero, no elements are undefined in this case. For zero inclination elements χ and ψ are zero and again no elements are undefined. Reformulating the state transition/coordinate state transformation using the equinoctal elements is recommended to eliminate the coplanar and circular/near circular orbit difficulties.

The equation for position and velocity for the satellite is initially computed using a perifocal coordinate system. These calculations are dependent upon an orbit where a perigee exists. Selection of a coordinate system not dependent upon the existence of a perigee will eliminate this problem. All computations in an inertial coordinate system is one way of solving this problem.

The difficulties experienced when the initial estimate of the relative element vector is perturbed from the true relative element vector are of primary concern since these difficulties would limit the scope with which the estimator could be used operationally. A good assumption is that the initial estimate used to start the operation of the estimator will come from a ground-based tracking facility and not be the true set of orbital elements due to limitations in the tracking accuracy. Starting with the initial estimate the interceptor, using range and range rate measurements, can estimate a more accurate set of orbital elements for the target. Presently a highly accurate initial estimate is required to insure convergence upon a solution. This highly accurate initial estimate may in fact be more accurate than the set the estimator could provide. Therefore, the use of the estimator would introduce more error to the orbit determination process. Reformulating or modifying the state transition/coordinate transformation matrix using the Equinoctal elements may increase estimator versatility. Adding relative angular measurements to supplement the range

and range rate measurements may also enhance the estimator's ability to successfully converge to a solution starting with a less accurate initial estimate of the element vector. Incorporating more types of observational data, such as angular measurements, would give more information to the estimator to process. Angular measurement may also increase the accuracy of the estimator's prediction of angular elements, thereby decreasing the variances of the angular elements to values comparable to the dimensional elements.

Modifying relative position and velocity estimators, such as the NASA Apollo Rendezvous Filter, with the state vector coordinate transformation technique developed in this design can create a new orbit determination method to be used by satellite tracking facilities from a space-based platform, such as the Space Shuttle or an orbiting station. The modified estimator could be designed and tested similarly to the method used in this study. The truth model would have to be modified to give relative angular measurements and a coordinate transformation from inertial to vehicle reference frame would be required.

Due to the requirement of a highly accurate initial estimate necessary to start the estimation process this design is not yet considered by me to be operational. However, with continued research and development of the ideas and techniques used in this study along with implementation of my recommendations can bring us closer to develop-

ment of an operational space based orbital estimation process to supplement and enhance our ground based tracking facilities in tracking and cataloging the many orbiting vehicles in space, both friendly and others.

Bibliography

1. Bate, Roger R., Mueller, Donald D., and White, Jerry E. Fundamentals of Astrodynamics. New York: Dover Publications, Inc., 1971.

2. Bowen, Douglas, Engineer, NORAD, Space Computational Center (Personal Correspondence and Technical Notes provided). Colorado Springs, Colorado, 15 October 81.

3. Devezin, Gerry, NASA Engineer (Personal Correspondence). Houston, Texas, 5 October 81.

4. Eades, J. B. and Drewry, J. W., "Relative Motion of Satellites in Near Orbits," Celestial Mechanics, 1973.

5. Kaplan, Marshall H., Modern Spacecraft Dynamics and Control. New York, John Wiley and Sons, 1976.

6. Leibold, A., "Orbit Determination Error Analysis Through Satellite-Satellite Systems," Proceedings of AOCS Conference. Noordiwijk, Germany, November 1977.

7. Muller, E. and Kachman, P., A New Approach to On-Board Orbit Navigation. Cambridge, Mass: MIT, Charles Stack Draper Laboratory, 1971.

8. Muller, Eugene, Professor, Massachusetts Institute of Technology (Personal Correspondence). Cambridge, Massachusetts, 6 October 81.

9. Weisel, William E., Lecture Materials Distributed in MC 636 and MC 731, School of Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1981.

Appendix A

Vector/Matrix Components of the Least Squares Estimation Equation

1. State Transition/State Coordinate Transformation Matrix, Φ

The Φ used in this design is not a classical Φ matrix. Φ relates the relative position and velocity state vector and the relative orbital element state vector. Φ also propagates the state vector in time. Φ is formed by developing relationships between the satellite position and velocity in the perifocal, PQW, coordinate frame and the Delaunay orbital elements and then transforming the result into the inertial, IJK, coordinate frame. Position and velocity are denoted here as $\overline{X}_{rv}$ and the orbital elements are denoted $\overline{X}_{ELM}$ and are defined in Equations (1) and (2).

$$\overline{X}_{rv} = [r_I,\ r_J,\ r_K,\ V_I,\ V_J,\ V_K] \tag{1}$$

$$\overline{X}_{ELM} = [L,\ l,\ G,\ g,\ H,\ h] \tag{2}$$

Equation (3) gives the PQW position and velocity relation to the elements.

$$\frac{d(\overline{X}_{rv})_{PQW}}{d(\overline{X}_{ELM})}_{6x6} = \frac{d(\overline{X}_{r,v})_{PQW}}{dE} \cdot \frac{dE}{d(\overline{X}_{ELM})} + \frac{\partial(\overline{X}_{r,v})_{PQW}}{\partial(\overline{X}_{ELM})} \tag{3}$$

Transforming Equation (3) into the IJK frame is accomplished by Equation (4).

$$\left[\frac{d(\bar{X}_{r,v})_{IJK}}{d(\bar{X}_{ELM})}\right] = \left[\frac{\partial R}{\partial(\bar{X}_{ELM})}\right] (\bar{X}_{r,v})_{PQW} + [R]\left[\frac{d(\bar{X}_{r,v})_{PQW}}{d\bar{X}_{ELM}}\right] \quad (4)$$

where

R = Rotation matrix PQW to IJK frame.

Φ is a 6x6 matrix given as Equation (5).

$$\Phi = \left[\frac{d(\bar{X}_{r,v})_{IJK}}{d(\bar{X}_{ELM})}\right] \quad (5)$$

The rotation matrix is a function of g, h, G, H. R is given in Equation (6) where C denotes cos and S denotes sin function. R transforms a position or velocity vector from the PQW to IJK coordinate frame.

$$[R] = \begin{bmatrix} C(h)C(g)-S(h)S(g)(H/G) & -C(h)S(g)-S(h)C(g)(H/G) & S(h)\sqrt{1-H^2/G^2} \\ S(h)C(g)+C(h)S(g)(H/G) & -S(h)S(g)+C(h)C(g)(H/G) & -C(h)\sqrt{1-H^2/G^2} \\ S(g)\sqrt{1-H^2/G^2} & C(g)\sqrt{1-H^2/G^2} & H/G \end{bmatrix}$$

(Equation 6)

Equations (7), (8), (9), and (10) give the partial derivative of R with respect to the Delaunay elements.

$$\left[\frac{\partial R}{\partial g}\right] = \begin{bmatrix} -C(h)S(g)+S(h)C(g)(H/G) & S(h)S(g)H/G-C(h)C(g) & 0 \\ -S(h)S(g)+C(h)C(g)(H/G) & -S(h)C(g)+C(h)S(g)(H/G) & 0 \\ C(g)\sqrt{1-H^2/G^2} & -S(g)\sqrt{1-H^2/G^2} & 0 \end{bmatrix}$$

(Equation 7)

$$\left[\frac{\partial R}{\partial h}\right]=\begin{bmatrix} -S(h)C(g)+(H/G)S(g)C(h) & S(h)S(g)-C(h)C(g)(H/G) & C(h)\sqrt{1-H^2/G^2} \\ C(h)C(g)-S(h)S(g)(H/G) & -C(h)S(g)+S(h)C(g)(H/G) & S(h)\sqrt{1-H^2/G^2} \\ 0 & 0 & 0 \end{bmatrix}$$

(Equation 8)

$$\left[\frac{\partial R}{\partial G}\right]=\begin{bmatrix} S(h)S(g)(H/G^2) & S(h)C(g)H/G^2 & S(h)(H^2/G^3)/\sqrt{1-H^2/G^2} \\ -C(h)S(g)H/G^2 & -C(h)C(g)H/G^2 & -C(h)(H^2/G^3)/\sqrt{1-H^2/G^2} \\ S(g)(H^2/G^3)/\sqrt{1-H^2/G^2} & C(g)(H^2/G^3)\sqrt{1-H^2/G^2} & -H/G^2 \end{bmatrix}$$

(Equation 9)

$$\left[\frac{\partial R}{\partial H}\right]=\begin{bmatrix} -S(h)S(g)/G & -S(h)C(g)/G & -S(h)(H/G^2)/\sqrt{1-H^2/G^2} \\ C(h)S(g)/G & C(h)C(g)/G & C(h)(H/G^2)/\sqrt{1-H^2/G^2} \\ -S(g)(H/G^2)/\sqrt{1-H^2/G^2} & -C(g)(H/G^2)/\sqrt{1-H^2/G^2} & 1/G \end{bmatrix}$$

(Equation 10)

The derivatives of the position and velocity in the PQW frame with respect to the eccentric anomaly are given in Equations (11) through (16).

$$dr_P/dE = (-L^2\mu)\ \sin(E-E_o) \qquad (11)$$

$$dr_Q/dE = (LG/\mu)\ \cos(E-E_o) \qquad (12)$$

$$dr_W/dE = 0 \qquad (13)$$

$$dV_p/dE = \frac{\mu L[\sin^2(E)\sqrt{1-G^2/L^2}\ -\cos(E)(1-\sqrt{1-G^2/L^2}\cos(E))]}{[L(1-\sqrt{1-G^2/L^2}\ \cos(E))]^2} \qquad (14)$$

$$dV_Q/dE = \frac{\mu L^2 G[\cos(E)\sin(E)\sqrt{1-G^2/L^2} - (1-\sqrt{1-G^2/L^2}\cos(E))\sin(E))]}{[L^2(1-\sqrt{1-G^2/L^2}\cos(E))]^2} \tag{15}$$

$$\frac{dV_W}{dE} = 0 \tag{16}$$

The partials of the position and velocity in the PQW frame with respect to the Delaunay Elements are given in Equations (17) through (29).

$$\frac{\partial r_P}{\partial L} = \frac{2L}{\mu}\cos(E) - \frac{2L}{\mu}\sqrt{1-G^2/L^2} - \frac{L^2}{\mu}\left(\frac{1}{\sqrt{1-G^2/L^2}}\right)(G^2/L^3) \tag{17}$$

$$\frac{\partial r_Q}{\partial L} = \frac{G}{\mu}\sin(E) \tag{18}$$

$$\frac{\partial r_W}{\partial L} = 0 \tag{19}$$

$$\frac{\partial V_P}{\partial L} = \frac{\mu\sin(E)[1-\sqrt{1-G^2/L^2}\cos(E) - (G^2/L^2)\cos(E)/\sqrt{1-G^2/L^2}]}{[L(1-\sqrt{1-G^2/L^2}\cos(E))]^2} \tag{20}$$

$$\frac{\partial V_Q}{\partial L} = \frac{\mu G\cos(E)[2L(1-\sqrt{1-G^2/L^2}\cos(E))-(G^2/L)\cos(E)/(\sqrt{1-G^2/L^2}]}{[L^2(1-\sqrt{1-G^2/L^2}\cos(E))]^2} \tag{21}$$

$$\frac{\partial V_W}{\partial L} = 0 \tag{22}$$

$$\frac{\partial r_P}{\partial G} = G/\left(\mu\sqrt{1-G^2/L^2}\right) \tag{23}$$

$$\frac{\partial r_Q}{\partial G} = \frac{L}{\mu}\sin(E) \tag{24}$$

$$\frac{\partial r_W}{\partial G} = 0 \tag{25}$$

$$\frac{\partial V_P}{\partial G} = \frac{[\mu G\sin(E)\cos(E)/(L\sqrt{1-G^2/L^2})]}{[L(1-\sqrt{1-G^2/L^2}\cos(E))]^2} \tag{26}$$

$$\frac{\partial V_Q}{\partial G} = \frac{v\cos(E)[L^2(1-\sqrt{1-G^2/L^2}\ \cos(E))-\cos(E)G^2/\sqrt{1-\ G^2/L^2}]}{[L^2(1-\sqrt{1-G^2/L^2}\cos(E))]^2} \tag{27}$$

$$\frac{\partial V_W}{\partial G} = 0 \tag{28}$$

$$\frac{\partial \bar{X}_{PQW}}{\partial_{l,g,h,H}} = 0 \tag{29}$$

The derivative of E with respect to L is given in Equation (30).

$$\frac{dE}{dL} = [(1/\sqrt{1-G^2/L^2})(G^2/L^3)\sin(E)]/[1-\sqrt{1-G^2/L^2}\ \cos(E)] \tag{30}$$

The derivative of E with respect to l is given in Equation (31).

$$\frac{dE}{dl} = [1/(1-\sqrt{1-G^2/L^2}\ \cos(E))] \tag{31}$$

The derivative of E with respect to G is given in Equation (32).

$$\frac{dE}{dG} = [(1/\sqrt{1-G^2/L^2})G\ \sin(E)/L^2]/[1-\sqrt{1-G^2/L^2}\ \cos(E)] \tag{32}$$

2. Observation Vector and Relation

Range and range rate are measured quantities, and each time these measurements are taken the following vectors and matrices are assembled. The observation vector Z, is given in Equation (34).

$$\overline{Z} = \begin{bmatrix} \text{Range to Target} \\ \\ \text{Range Rate} \end{bmatrix} \tag{33}$$

$\overline{Z}$ is a function of the relative position and velocity, as shown in Equation (34).

$$\overline{Z} = \begin{bmatrix} (\Delta r_I{}^2 + \Delta r_J{}^2 + \Delta r_K{}^2)^{1/2} \\ (\Delta V_I \Delta r_I + \Delta V_J \Delta r_J + \Delta V_K \Delta r_K)/\text{RANGE} \end{bmatrix} \tag{34}$$

If the relative position and velocity are defined in vector $\overline{X}_{IJK}$ then $\overline{Z}$ is given as Equation (35) where the observation vector is a nonlinear function called the observation relation G.

$$\overline{Z} = G(\overline{X},t) \tag{35}$$

Relating the state vector $\overline{X}_{IJK}$ to the observation vector Z is accomplished using Equation (36).

$$H = \frac{\partial G(\overline{X},t)}{\partial \overline{X}} \tag{36}$$

In detail H is given in Equation (37).

$$H = \begin{bmatrix} \frac{\partial RANGE}{\partial \Delta r_I} & \frac{\partial RANGE}{\partial \Delta r_J} & \cdots\cdots & \frac{\partial RANGE}{\partial \Delta V_K} \\ \frac{\partial RANGE\ RATE}{\partial \Delta r_I} & & \cdots\cdots\cdots & \frac{\partial RANGE\ RATE}{\partial \Delta V_k} \end{bmatrix} \quad (37)$$

where the components of H are given in Equations (38) through (40). Note that range is not a function of relative velocity.

$$\frac{\partial RANGE}{\partial \Delta r_{I,J,K}} = \frac{\Delta r_{I,J,K}}{RANGE} \quad (38)$$

$$\frac{\partial RANGE\ RATE}{\partial \Delta r_{I,J,K}} = \frac{(RANGE)\Delta V_{I,J,K} - (\Delta V \cdot \Delta r)\Delta r_{I,J,K}/RANGE}{RANGE^2} \quad (39)$$

$$\frac{\partial RANGE\ RATE}{\partial \Delta V_{I,J,K}} = \frac{\Delta r_{I,J,K}}{RANGE} \quad (40)$$

$\mathcal{H}$ is given by Equation (41).

$$\mathcal{H} = H\ \Phi \quad (41)$$

so in detail $\mathcal{H}$ is represented by Eq (42).

$$\mathcal{H} = \begin{bmatrix} \frac{dRANGE}{dL} & \frac{dRANGE}{dl} & \cdots\cdots & \frac{dRANGE}{dh} \\ \frac{dRANGE\ RATE}{DL} & & \cdots\cdots & \frac{dRANGE\ RATE}{dh} \end{bmatrix} \quad (42)$$

3. Q Matrix

The Q matrix is a diagnol matrix of the σ^2 values for the measurement device. 1 σ_{RANGE} = 100 meters and 1 $\sigma_{RANGE\ RATE}$ = .3 meters/second were used for the estimator radar. The size of the Q matrix is determined by the number of measurements taken, it is a 2Nx2N matrix where N is the number of measurements taken (Equation 43).

$$Q = \begin{bmatrix} \sigma^2_{R,1} & & & & & \\ & \sigma^2_{RR,1} & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \sigma^2_{R,N} & \\ & & & & & \sigma^2_{RR,N} \end{bmatrix} \qquad (43)$$

4. <u>Residual Vector</u>

A residual is defined as the difference between the measured value and the estimation of the measurement Equation (44).

$$\bar{r}_{1,N} = \begin{bmatrix} \text{RANGE 1} \\ \text{RANGE RATE 1} \\ \cdot \\ \cdot \\ \cdot \\ \text{RANGE N} \\ \text{RANGE RATE N} \end{bmatrix} - \left[G(X_{1,N}, t_{1,N}) \right]$$

where

$\bar{X}$ is the relative position and velocity vector.

5. <u>Nonlinear Least Squares Equation</u>

When the components thus far computed are substituted into the nonlinear least squares equation, Equation (45), the result is Equation (46), correction to the reference

relative orbital elements.

$$\delta\bar{X} = (\mathcal{H}^{T}_{1,N}\, Q^{-1}_{N}\, \mathcal{H}_{1,N})^{-1}\, \mathcal{H}^{T}_{1,N}\, Q^{-1}_{N}\, \bar{r}_{1,N} \tag{45}$$

$$\delta\bar{X} = \begin{bmatrix} \delta L \\ \delta l \\ \delta G \\ \delta g \\ \delta H \\ \delta h \end{bmatrix} \tag{46}$$

The target orbital elements are found using Equation (47).

$$\bar{X}_{ELEMENTS,TARGET} = \bar{X}_{ELEMENTS,INTERCEPTOR} + \delta\bar{X} \tag{47}$$

Appendix B

Test Case Data

Test case data is generated using the truth model computer program contained in Appendix C.

Test Case 1 Data

Classical Orbital Elements

Element	Interceptor	Target
a	1.3 DU	1.35 DU
e	0.2	0.3
i	45°	46°
ω	45°	46°
Ω	45°	46°
M_O	0°	0°

Delaunay Orbital Elements
Target - Interceptor

Δ Element	Value	
ΔL	0.02171957876309	DU^2/TU
Δl	0°	RAD
ΔG	-0.008762011386843	DU^2/TU
Δg	0.01745329252222	RAD
ΔH	-0.01999321219953	DU^2/TU
Δh	0.01745329252222	RAD

Tracking Target From Interceptor Over 1 Orbit at 10 Minute Intervals, No Noise Corruption On Data

Range (DU)	Range Rate (DU/TU)	Time (TU)	(MIN)
0.17513109	0.09800249	0.74366638	10
0.21953387	0.02029584	1.4873328	20
0.211625212	-0.03500555	2.23099914	30
0.17893223	-0.04364209	2.97466552	40
0.16317941	0.01209514	3.71833189	50
0.20293606	0.09071542	4.46199827	60
0.28797655	0.13177235	5.20566465	70
0.39208177	0.14477618	5.94933103	80
0.49886571	0.13946822	6.69299741	90
0.59442894	0.11326908	7.43666379	100
0.65887875	0.05266586	8.18033017	110
0.65981660	-0.05861977	8.92399655	120
0.567448959	-0.18229518	9.66766293	130

* Conversions
6378145 m = 1 DU
7.0905376 m/s = 1 DU/TU
806.8136 = 1 TU

Test Case 2 Data

Classical Orbital Elements

Element	Interceptor	Target
a	1.3 DU	1.35 DU
e	0.2	0.3
i	45°	46°
ω	45°	46°
Ω	45°	46°
M_O	0	0.1 RAD

Delaunay Orbital Elements
Target - Interceptor

Δ Element	Value
ΔL	0.02171957876309 DU^2/TU
Δl	0.1 RAD
ΔG	-0.008762011386843 DU^2/TU
Δg	0.01745329252222 RAD
ΔH	-0.01999321219953 DU^2/TU
Δh	0.01745329252222 RAD

Tracking Target From Interceptor Over 1 Orbit at 10 Minute Intervals Between Data, No Noise On Data

Range (DU)	Range Rate (DU/TU)	Time (TU)	(MIN)
0.33780912	0.08567198	0.74366638	10
0.36540682	-0.00667171	1.48733276	20
0.337044592	-0.06346114	2.23099914	30
0.27953501	-0.08556216	2.97466552	40
0.21942767	-0.06804477	3.71833189	50
0.192346278	0.00356536	4.46199827	60
0.22684532	0.083173361	5.20566465	70
0.30400591	0.117095657	5.94933103	80
0.392584098	0.116469891	6.69299741	90
0.47041812	0.08780420	7.43666379	100
0.514020451	0.02200318	8.18033017	110
0.491439874	-0.08888904	8.92399655	120
0.384083659	-0.18700137	9.66766293	130

* Conversions
6378145 m = 1 DU
7.0905376 m/s = 1 DU/TU
806.8136 = 1 TU

Test Case 3 Data

Classical Orbital Elements

Element	Interceptor	Target
a	1.3 DU	1.35 DU
e	0.2	0.3
i	45°	46°
ω	45°	46°
Ω	45°	46°
M_O	0	0.1 RAD

Delaunay Orbital Elements
Target - Interceptor

Δ Element	Value	
ΔL	0.02171957876309	DU^2/TU
Δl	0.1	RAD
ΔG	-0.008762011386843	DU^2/TU
Δg	0.01745329252222	RAD
ΔH	-0.01999321219953	DU^2/TU
Δh	0.01745329252222	RAD

Tracking Target From Interceptor Over 1 Orbit at 10 Minute Intervals Between Data. Two Batches of Three Sequential Measurements Follow Initial Set

Range (DU)	Range Rate (DU/TU)	Time (TU)	(MIN)
Batch 1			
0.24317394	-0.173231099	10.4113293	140
0.1431328	-0.08826717	11.15499568	150
0.124736178	+0.041287686	11.89866206	160
Batch 2			
0.18910942	0.11915443	12.64232844	170
0.29104299	0.15135906	13.38599482	180
0.41097969	0.16973908	14.1296612	190

Noise Corruption

Noise corruption to the data is determined by random number generator that determines the position along a square noise function modelled with a 1σ value equivalent to a Gaussian σ value. The square function is superimposed over the Gaussian distribution. The equivalent 1σ value is found by equating the expressions for σ^2 for the gaussian and square probability functions and finding the square σ in terms of the Gaussian function σ. This is shown below.

$$(1) \quad \sigma^2_{square} = \int_{-W_{Square}}^{W} x^2 \frac{1}{2W} \, dx$$

$$\sigma^2_{Gaussian} = \int_{-\infty_{Gaussian}}^{\infty} x^2 \frac{1}{\sqrt{2\pi}\,\sigma_o} e^{-x^2/\sigma^2} dx$$

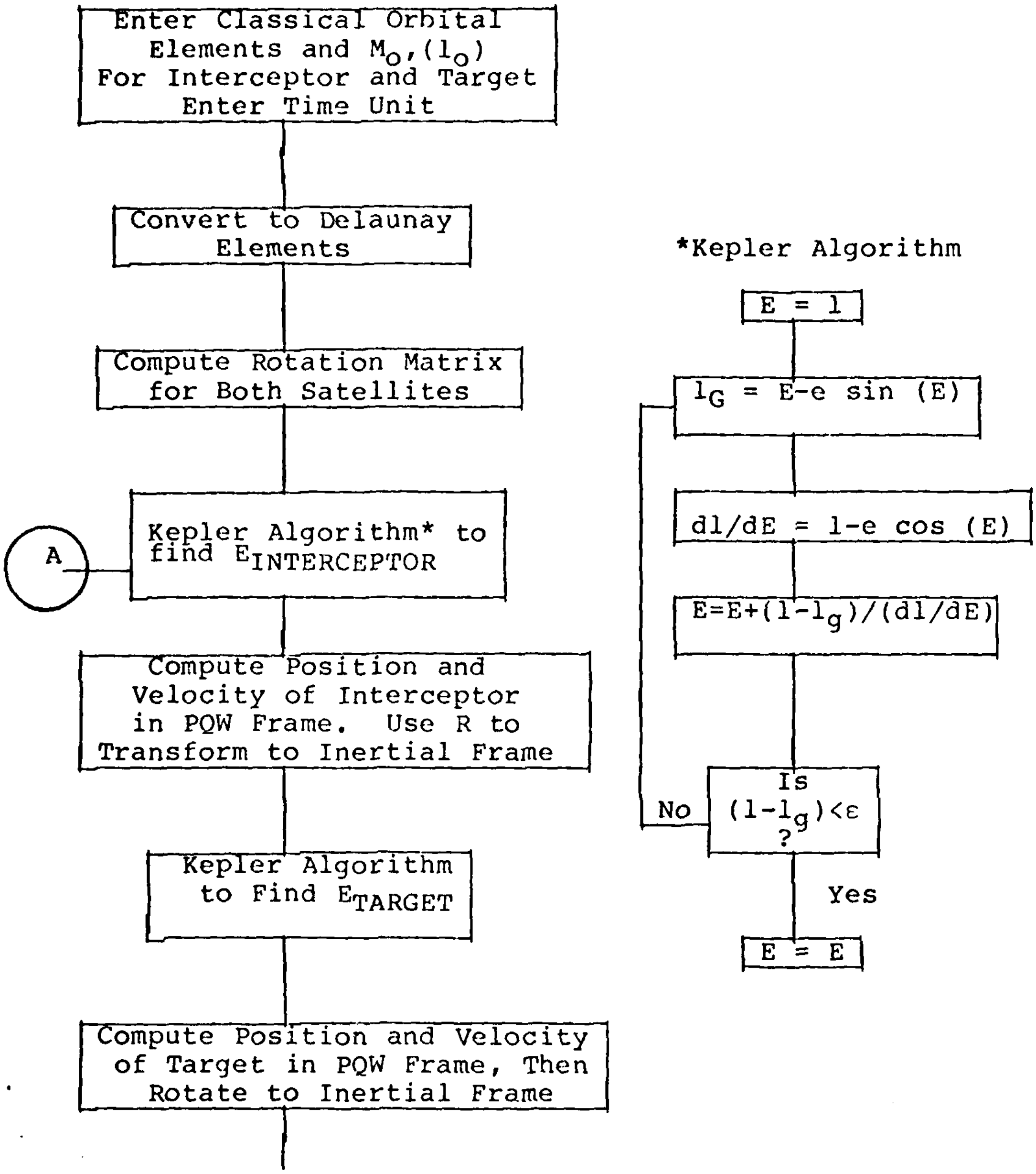
$$\sigma^2_{Gaussian} = \frac{1}{3} x^3 \left(\frac{1}{2W}\right) \Bigg|_{-W}^{W} = \frac{2W^3}{6W} = \frac{W^2}{3}$$

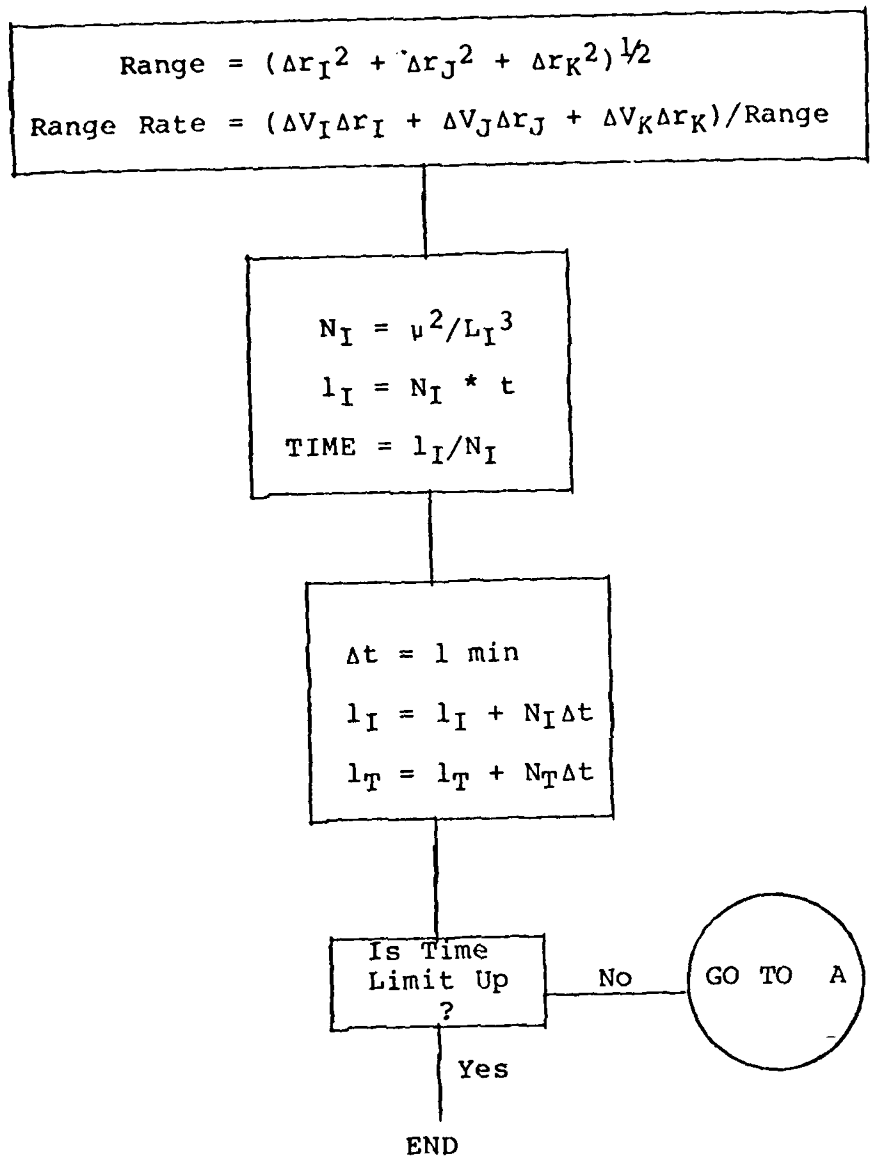
So the square function σ, called W, is equivalent to $\sqrt{3}\sigma$ Gaussian, Equation (2).

$$(2) \quad W = \sigma_{Square} = \sqrt{3}\sigma_{Gaussian}$$

Appendix C

Truth Model Flowchart and Computer Program




Range $= (\Delta r_I^2 + \Delta r_J^2 + \Delta r_K^2)^{1/2}$
Range Rate $= (\Delta V_I \Delta r_I + \Delta V_J \Delta r_J + \Delta V_K \Delta r_K)/$Range
$N_I = \mu^2/L_I^3$
$l_I = N_I * t$
TIME $= l_I/N_I$
$\Delta t = 1$ min
$l_I = l_I + N_I \Delta t$
$l_T = l_T + N_T \Delta t$
Is Time Limit Up ?
No
GO TO A
Yes
END

TRUTH MODEL COMPUTER PROGRAM

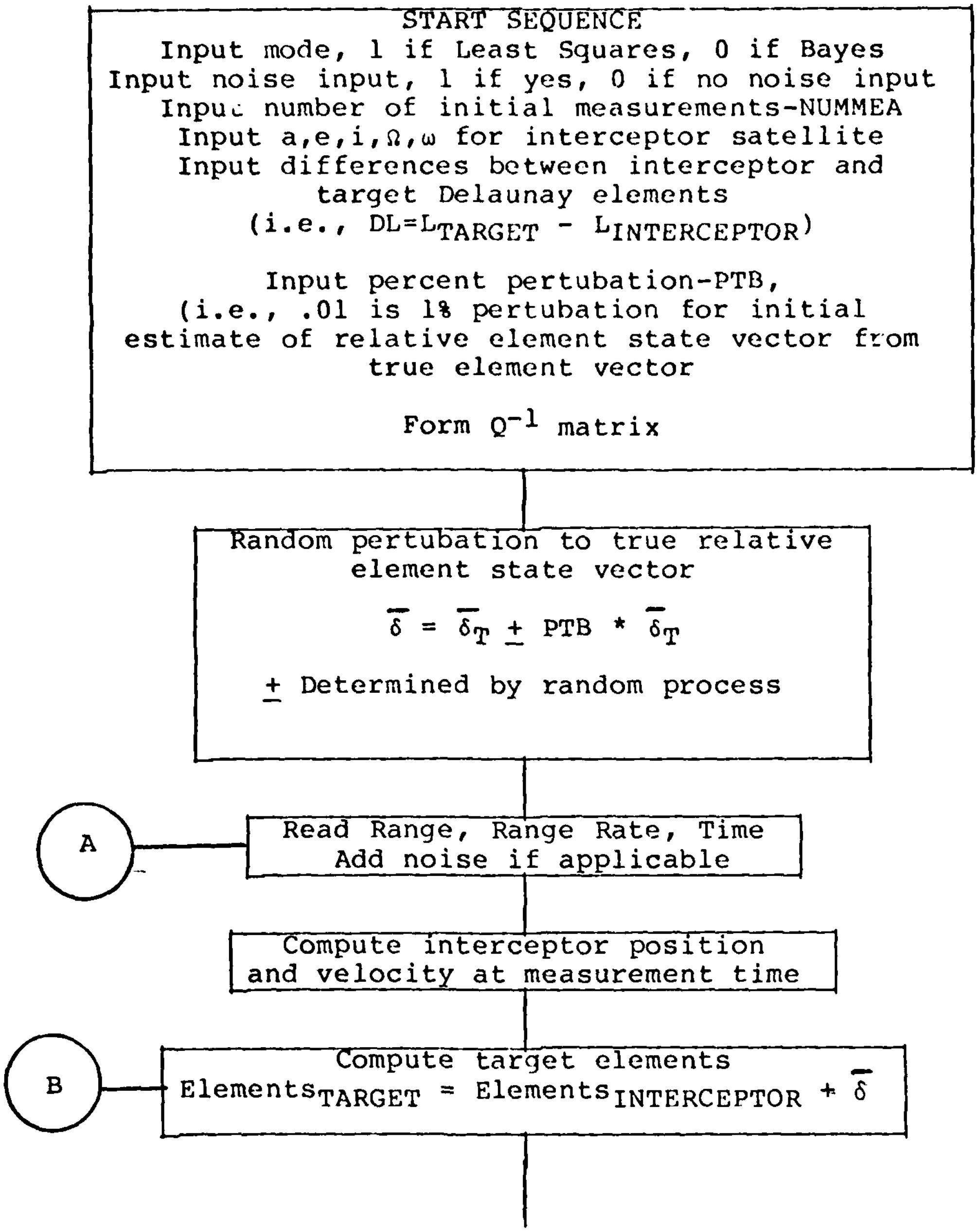
```
***    TRUTH MODEL PROGRAM '

PROGRAM DATAG
 REAL LI,GI,HI,GSI,HSI,LSI,LT,GT,HT,GST,HST,LST,NI,NT,
*EI,ET,TEST,LSG,DMDE,HS,GS,H,G,TAU,DELT,DATE,TIME,DELI,TPPI,TPPT,
*AXI,AXT,ECCI,ECCT,LNODI,LNODT,ARGPI,ARGPT,INCLI,INCLT,PI,TWOPI,
*TIMMIN
 REAL POS(3,1),VEL(3,1),ROTT(3,3),POSI(3,1),VELI(3,1),POST(3,1),
*VELT(3,1),ROTT(3,3),POSTI(3,1),VELTI(3,1),PREL(3,1),VREL(3,1)
 INTEGER CYCLE,CNT,LIMIT
 COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,M
 CYCLE=
***    SET TIME LIMIT FOR TRACKING     **
 PRINT *,' ENTER TIME REQUIRED FOR TRACKING '
 READ *,LIMIT
 PI=3.14159265
 TWOPI=2*PI
 DELT= .7435E-375
 CNT=-1
***    INPUT ORBITAL ELEMENTS FOR INTERCEPTOR AND TARGET    ***
 PRINT *,' INPUT CLASSICAL ELEMENTS OF INTERCEPTOR AND DELAUNAY LS'
 PRINT *,' IN ORDER, AX,ECC,LNOD,ARGP,INCL,LS   '
 READ *,AXI,ECCI,LNODI,ARGPI,INCLI,LSI
 PRINT *,' INPUT CLASSICAL ELEMENTS OF TARGET AND DELAUNAY LS '
 PRINT *,' IN ORDER, AX,ECC,LNOD,ARGP,INCL,LS        '
 READ *,AXT,ECCT,LNODT,ARGPT,INCLT,LST
 PRINT *,' CLASSICAL ORBITAL ELEMENTS OF TARGET AND INTERCEPTOR '
 PRINT *,'      INTERCEPTOR ELEMENTS '
 PRINT *,' '
 PRINT *,' AX= ',AXI,' ECC= ',ECCI,' LNODE = ',LNODI
 PRINT *,' ARGP= ',ARGPI,' INCL =',INCLI,' LSI = ',LSI
 PRINT *,' '
 PRINT *,'       TARGET ELEMENTS '
 PRINT *,' AX= ',AXT,' ECC= ',ECCT,' LNODE = ',LNODT
 PRINT *,' ARGP= ',ARGPT,' INCL =',INCLT,' LST = ',LST
 PRINT *,' '
***    COMPUTE DELAUNAY ELEMENTS    ***
 GSI=ARGPI*PI/180.
 GST=ARGPT*PI/180.
 HSI=LNODI*PI/180.
 HST=LNODT*PI/180.
 LI=SQRT(AXI)
 LT=SQRT(AXT)
 GI=LI*((1.-ECCI**2)**.5)
 GT=LT*((1.-ECCT**2)**.5)
 INCLI=INCLI*PI/180.
 INCLT=INCLT*PI/180.
 HI=GI*COS(INCLI)
 HT=GT*COS(INCLT)
 PRINT *,' '
 PRINT *,' INTERCEPTOR ELEMENTS  '
 PRINT *,' LI= ',LI,' GI= ',GI,' HI= ',HI
 PRINT *,'LSI= ',LSI,' GSI = ',GSI,' HSI= ',HSI
 PRINT *,' '
 PRINT *,' TARGET ELEMENTS '
 PRINT *,' LT= ',LT,' GT= ',GT,' HT= ',HT
 PRINT *,'LST= ',LST,' GST = ',GST,' HST= ',HST
 PRINT *,' '
 PRINT *,' DELTA ORBITAL ELEMENTS '
 PRINT *,' '
 PRINT *,' DELTA L = ',LT-LI
 PRINT *,' DELTA LS = ',LST-LSI
 PRINT *,' DELTA G = ',GT-GI
```

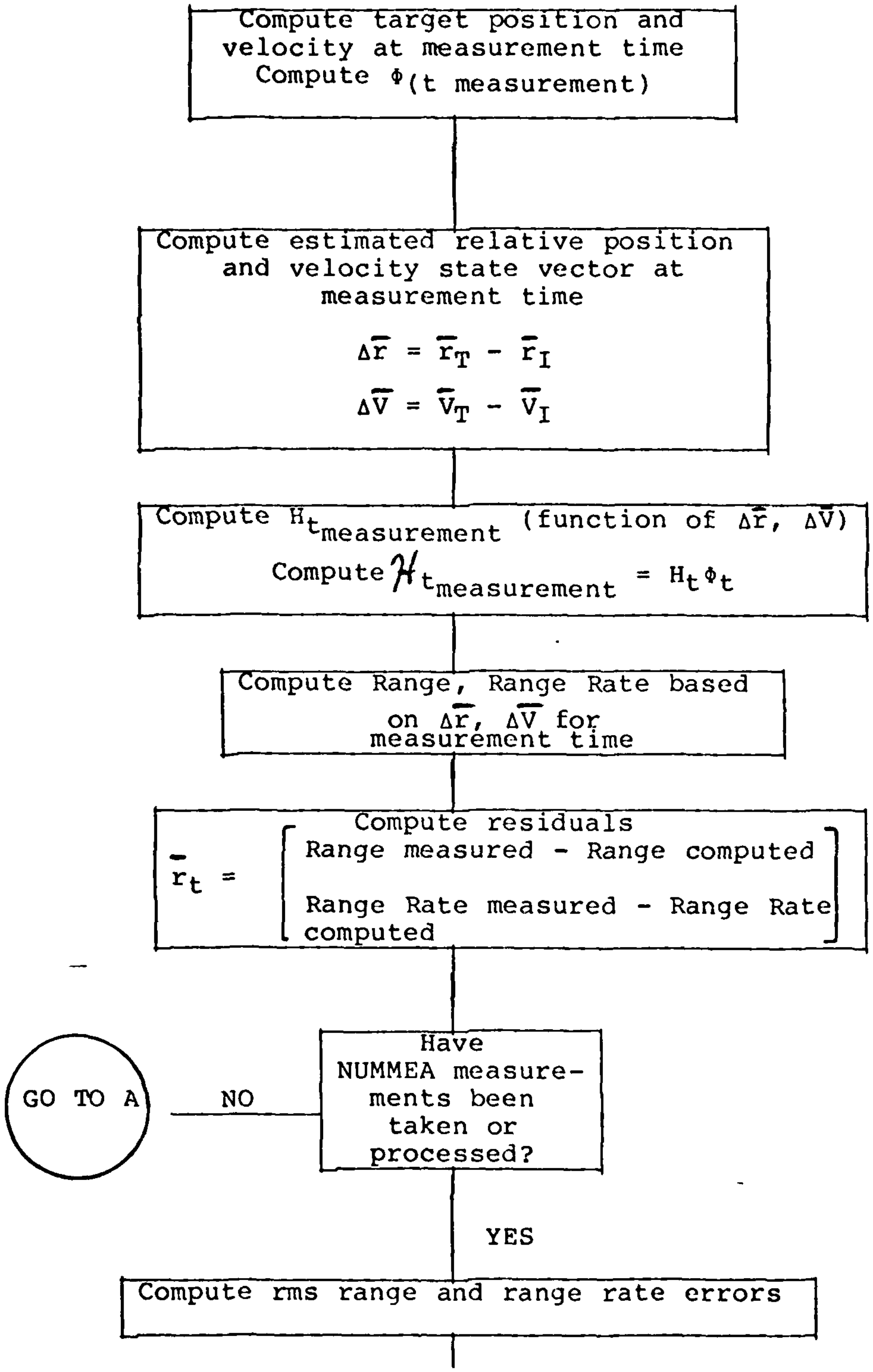
```
      PRINT *,' DELTA GI = ',G T-GSI
      PRINT  ,' DELTA H = ',HT-HI
      PRINT *,' DELTA HS = ',H T-HSI
      PRINT *,' '
      HS=HSI
      GS=GSI
      H=HI
      G=GI
      CALL ROT33(ROTI)
      HS=HST
      GS=GST
      H=HT
      G=GT
      CALL ROT33(ROTT)
      NI=1./LI **3
      NT=1./LT**3
5     CONTINUE
      CYCLE=CYCLE+1
***   COMPUTE INTERCEPTOR POSITION AND VELOCITY   ***
      EI=LSI
10    CONTINUE
      LSG=EI-ECCI*SIN(EI)
      DMDE=1.-ECCI*COS(EI)
      EI=EI+(LSI-LSG)/DMDE
      TEST=LSI-LSG
      TEST=ABS(TEST)
      IF (TEST .LT.1E-1 ) GO TO 20
      GO TO 10
20    CONTINUE
      POS(1,1)= (LI**2)*(COS(EI)-ECCI)
      POS(2,1)=(LI*GI)*SIN(EI)
      POS(3,1)=0.
      VEL(1,1)=(-1) SIN(EI)/(LI*(1.-ECCI*COS(EI)))
      VEL(2,1)=GI*COS(EI)/((LI**2) (1.-ECCI*COS(EI)))
      VEL(3,1)=0.
      CALL MULTI(ROTI,POS,POSI,3)
      CALL MULTI(ROTI,VEL,VELI,3)
***   COMPUTE TARGET POSITION AND VELOCITY   ***
      ET=LST
30    CONTINUE
      LSG=ET-ECCT*SIN(ET)
      DMDE=1.-ECCT*COS(ET)
      ET=ET+(LST-LSG)/DMDE
      TEST=LST-LSG
      TEST=ABS(TEST)
      IF (TEST .LT. 1E-1 ) GO TO 40
      GO TO 30
40    CONTINUE
      POST(1,1)=(LT**2)*(COS(ET)-ECCT)
      POST(2,1)=(LT*GT)*SIN(ET)
      POST(3,1)= .
      VELT(1,1)=(-1)*SIN(ET)/(LT*(1.-ECCT*COS(ET)))
      VELT(2,1)=GT*COS(ET)/((L **2)*(1.-ECCT*COS(ET)))
      VELT(3,1)=0.
      CALL MULTI(ROTT,POST,POSTI,3)
      CALL MULTI(ROTT,VELT,VELTI,3)
***   COMPUTE RELATIVE POSITION AND VELOCITY   ***
      DO 50 I=1,3
        PREL(I,1)=POSTI(I,1)-POSI(I,1)
        VREL(I,1)=VELTI(I,1)-VELI(I,1)
50    CONTINUE
***   COMPUTE RANGE AND RANGE RATE MEASUREMENT FOR TIME   ***
      RANGE=SQRT(PREL(1,1)**2+PREL(2,1)**2+PREL(3,1)**2)
      RRATE=(VREL(1,1)*PREL(1,1)+VREL(2,1)*PREL(2,1)+VREL(3,1)*PREL(3,1)
     *)/RANGE
      TIME=LSI/NI
```

```
      TIMMIN=TIME*(1./60.)
      CNT=CNT+1
      IF (CNT .EQ. 1 ) THEN
      PRINT *,' '
      PRINT '(2X,F7.2,3X,F13.9,3X,F13.9,3X,F13.9)',TIMMIN,TIME,
     *RANGE,FRATE
      CNT=0
      ENDIF
C***    TIME STEP ONE MINUTE AND GO BACK TO REPEAT    ***
      LSI=LSI+NI*DELT
      LST=LST+NT*DELT
      IF (CYCLE .GT. LIMIT) GO TO 99
      GO TO 5
99    END
```

Appendix D

Relative Orbital Element Estimator Flowchart and Computer Program




Compute target position and velocity at measurement time
Compute Φ(t measurement)
Compute estimated relative position and velocity state vector at measurement time
$\Delta\bar{r} = \bar{r}_T - \bar{r}_I$
$\Delta\bar{V} = \bar{V}_T - \bar{V}_I$
Compute $H_{t_{measurement}}$ (function of $\Delta\bar{r}$, $\Delta\bar{V}$)
Compute $\mathcal{H}_{t_{measurement}} = H_t \Phi_t$
Compute Range, Range Rate based on $\Delta\bar{r}$, $\Delta\bar{V}$ for measurement time
Compute residuals
$\bar{r}_t$ =
Range measured - Range computed
Range Rate measured - Range Rate computed
Have NUMMEA measurements been taken or processed?
NO
GO TO A
YES
Compute rms range and range rate errors

Compute $\Delta\delta$ (changes to the relative element state vector) using least squares estimation

$$\Delta\bar{\delta} = (\mathcal{H}^T Q^{-1} \mathcal{H})^{-1} \mathcal{H}^T Q^{-1} \bar{r}$$

Store $(\mathcal{H}^T Q^{-1} \mathcal{H})$ as $P^{-1}(-)$ for subsequent Bayes filter

Update relative element state vector

$$\bar{\delta}^{+} = \bar{\delta}^{-} + \Delta\bar{\delta}$$

Recompute target elements based on δ^{+}

$$\text{Elements}_T = \text{Elements}_I + \bar{\delta}^{+}$$

Recompute target position and velocity, based on updated elements for each measurement time

Recompute Δr, ΔV, range, range rate for each measurement time based on updated elements

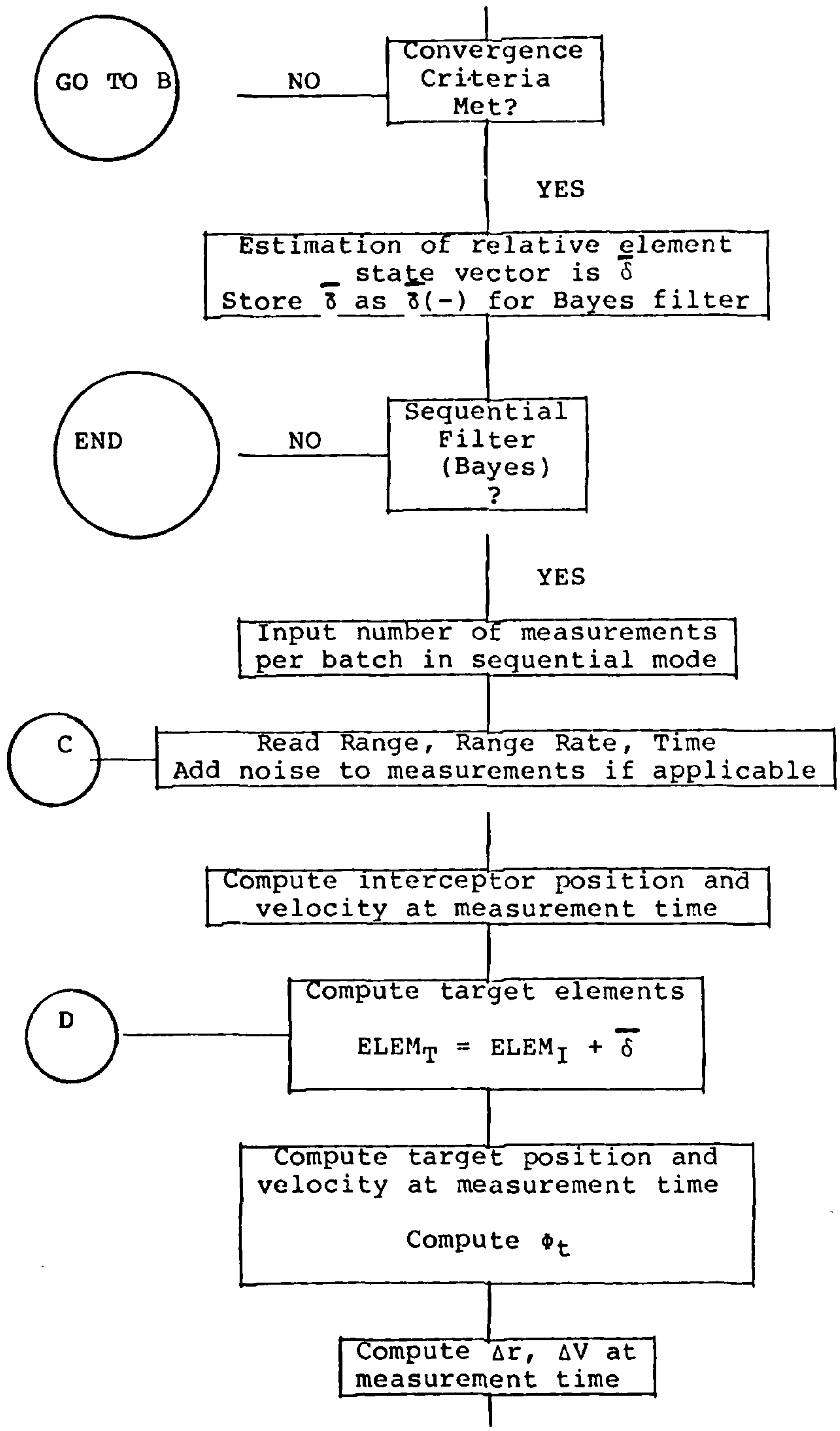
Compute residuals based on updated elements
Compute rms error values

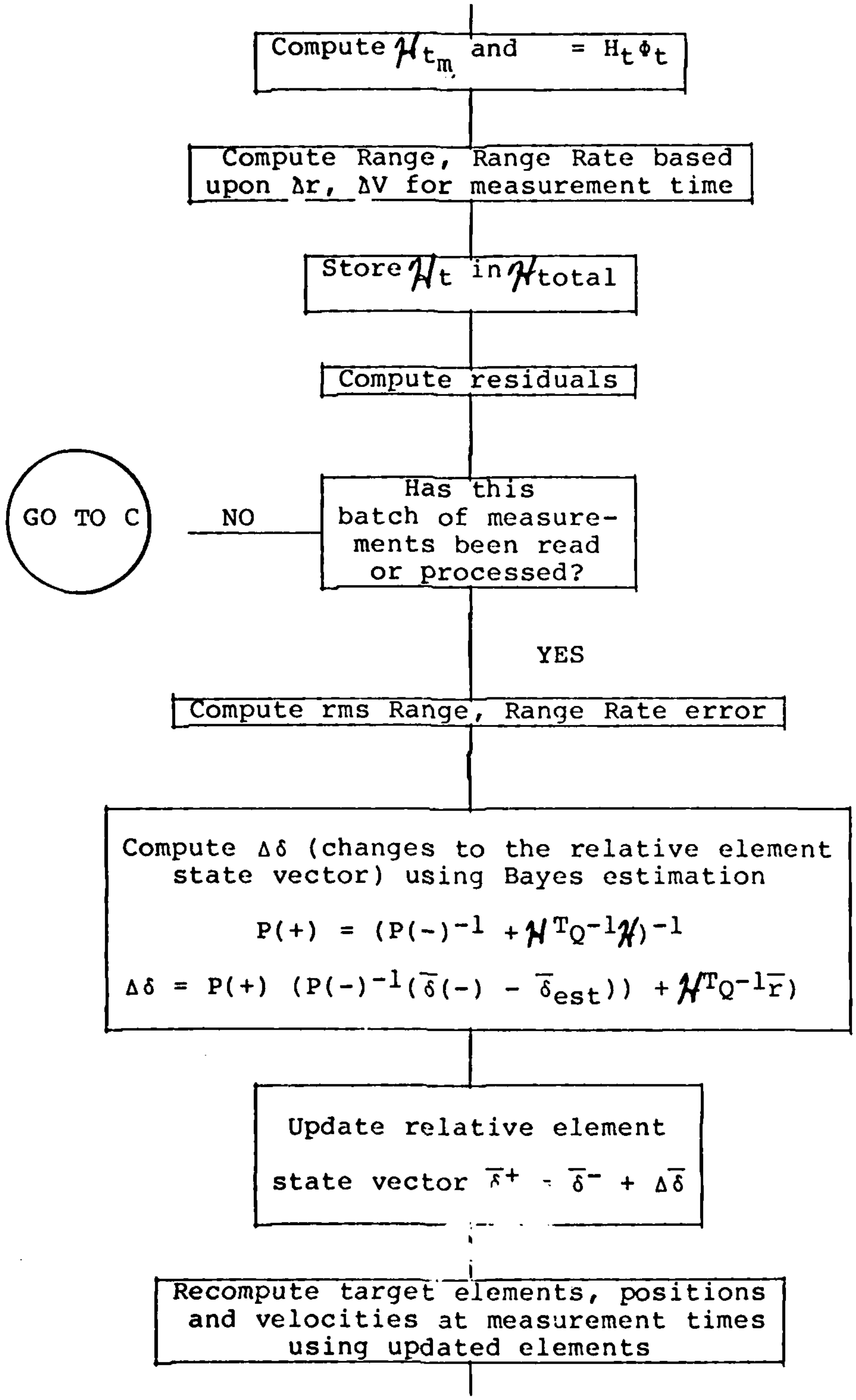
Convergence Check

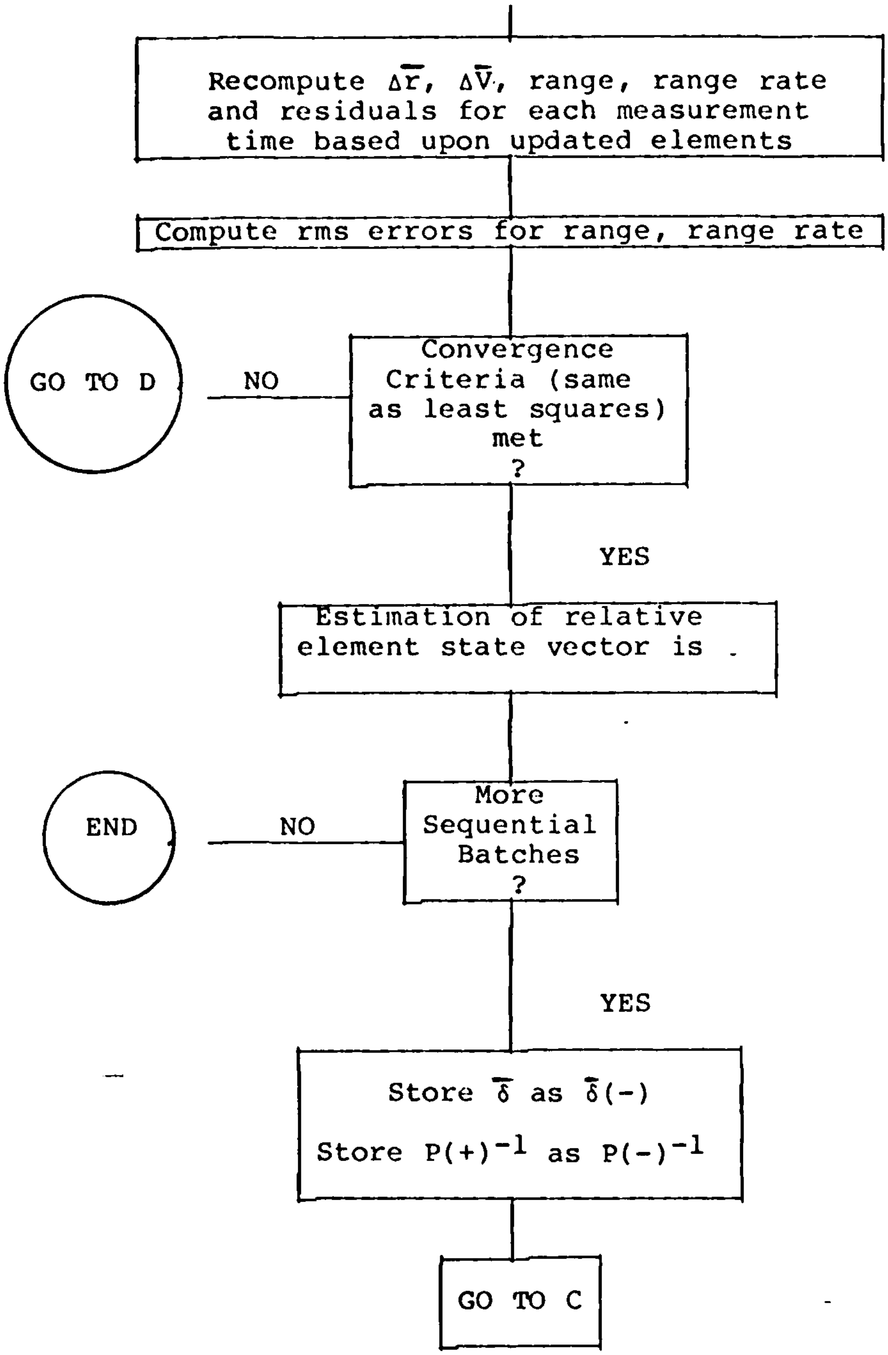
Range residuals $< 3\sigma_{range}$

Range rate residuals $< 3\sigma_{range\ rate}$

Monitor $\Delta\delta_i < \sqrt{P_{ii}}$


Convergence Criteria Met?
GO TO B
NO
YES
Estimation of relative element state vector is $\bar{\delta}$
Store $\bar{\delta}$ as $\bar{\delta}(-)$ for Bayes filter
Sequential Filter (Bayes) ?
END
NO
YES
Input number of measurements per batch in sequential mode
C
Read Range, Range Rate, Time
Add noise to measurements if applicable
Compute interceptor position and velocity at measurement time
D
Compute target elements
$ELEM_T = ELEM_I + \bar{\delta}$
Compute target position and velocity at measurement time
Compute Φ_t
Compute Δr, ΔV at measurement time


Compute $\mathcal{H}_{t_m}$ and $= H_t \Phi_t$
Compute Range, Range Rate based upon Δr, ΔV for measurement time
Store $\mathcal{H}_t$ in $\mathcal{H}_{total}$
Compute residuals
GO TO C
NO
Has this batch of measurements been read or processed?
YES
Compute rms Range, Range Rate error
Compute $\Delta\delta$ (changes to the relative element state vector) using Bayes estimation
$P(+) = (P(-)^{-1} + \mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$
$\Delta\delta = P(+)\ (P(-)^{-1}(\bar{\delta}(-) - \bar{\delta}_{est})) + \mathcal{H}^T Q^{-1} \bar{r})$
Update relative element state vector $\bar{\delta}^+ = \bar{\delta}^- + \Delta\bar{\delta}$
Recompute target elements, positions and velocities at measurement times using updated elements


Recompute $\Delta\bar{r}$, $\Delta\bar{V}$, range, range rate and residuals for each measurement time based upon updated elements
Compute rms errors for range, range rate
GO TO D
NO
Convergence Criteria (same as least squares) met ?
YES
Estimation of relative element state vector is .
END
NO
More Sequential Batches ?
YES
Store $\bar{\delta}$ as $\bar{\delta}(-)$
Store $P(+)^{-1}$ as $P(-)^{-1}$
GO TO C

DEFINITIONS OF VARIABLES USED IN MAIN PROGRAM

Real Variables

AX	Semi major axis of interceptor (DU)
ARGP	Argument of perigee of interceptor (deg, RAD)
CK	Difference between the computed and actual mean anomaly in the Kepler Algorithm
C1-C6	Diagnol elements of the Covariance matrix
DXPDE	Dr_P/DE, derivative of the position vector component along the perigee direction with respect to eccentric anomaly
DXQDE	Dr_Q/DE
DXWDE	Dr_W/DE
DVPDE	DV_P/DE
DVQDE	DV_Q/DE
DVWDE	DV_W/DE
DEDL	DE/DL, Derivative of the eccentric anomaly with respect to the Delaunay element L
DEDLS	DE/Dl
DEDG	DE/DG
DLSCDE	Dl/DE, used in Kepler algorithm
DL	$L_{target} - L_{interceptor}$
DLS	$l_{target} - l_{interceptor}$
DG, DGS, DH, DHS	See above
E, ECEN, ECENT	Eccentric anomaly
ECC	Eccentricity interceptor
ECCTGT	Eccentricity target

G	Delaunay element G target
GI	Delaunay element G interceptor
GS	Delaunay elemeng g target
GSI	Delaunay element g interceptor
H, HI, HS, HSI	See above
INCL	Inclination of interceptor orbit
L	Element L for target
LI	Element L for interceptor
LS, LST	Element l target
LSI	Element l interceptor
LSC	l computed in Kepler algorithm
LNOD	Longitude of ascending node of interceptor
MEANMO	Mean motion of orbit
MU	Gravitational constant, 1 DU^3/TU^2
NOSR	Noise for range
NOSRR	Noise for range rate
PTB	Pertubation input for elements
PI	3.141592654
RANGE	Range
RRATE	Range rate
RANGEG	Range computed by estimator
RRATEG	Range rate computed by estimator
RKM	Range converted to KM
RRKM	Range rate converted to KM/sec
SIGR	Gaussian σ for range measurement
SIGRR	Gaussian σ for range rate measurement

TIME	Time in TU
TIMEM	Time in minutes
VPPL	$\partial V_P/\partial L$
VQPL	$\partial V_Q/\partial L$
VWPL	$\partial V_W/\partial L$
VPPG	$\partial V_P/\partial G$
VQPG	$\partial V_Q/\partial G$
XPPL	$\partial r_P/\partial L$
XQPL	$\partial r_Q/\partial L$
XWPL	$\partial r_W/\partial L$
XPPG	$\partial r_P/\partial G$
XQPG	$\partial r_Q/\partial G$

<u>Integer Variables</u>

B1	Number of measurements x 2
B2	B1 - 1
C10	Counter
COUNT	Counter
CYCLE	Counter
IER	Error parameter for MATRIX inversion subroutine
J	Counter
K	Counter
ID, ID1	Measurement of identifiers
MODE	Mode of program, 1 if batch, 0 if sequential
NUMMEA	Number of measurements to be taken

NOSE	Noise parameter, 1 if noise corruption to data
N	Counter
NN	Counter
SEED	Start parameter for random number generator
STORE	Counter
RR	Counter
XX	Counter

<u>Arrays</u> (<u>Vectors</u> <u>and</u> <u>Matrices</u>)

PRPG	$[\partial R/\partial G]_{6x6}$
PRPGS	$[\partial R/\partial g]_{6x6}$
PRPH	$[\partial R/\partial H]_{6x6}$
PRPHS	$[\partial R/\partial h]_{6x6}$
XPQW	Position and velocity vector in PQW frame, $\bar{X}_{6x1}$
R	Rotation matrix PQW → IJK, $[R]_{6x6}$
DXDL	$[dX/dL]_{6x1}$
DXDLS	$[dX/dl]_{6x1}$
DXDG	$[dX/dG]_{6x1}$
XIDL	$[dX_{INERTIAL}/dL]_{6x1}$ $\bar{X}_{INERTIAL}$ is $\bar{X}$ rotated to IJK frame
XIDLS	$[dX_I/dl]_{6x1}$
XIDG	$[dX_I/dG]_{6x1}$
IDGS	$[dX_I/dg]_{6x1}$
XIDH	$[dX_I/dH]_{6x1}$
[illegible]	$[dX_I/dh]_{6x1}$

TRANS	Φ matrix, composed of column vectors $dX_I/dL, l, G, g, H, h$. This is the state transition/coordinate transformation matrix Φ
DEL	Vector of the differences between the target and interceptor elements $[\Delta L, \Delta l, \Delta G, \Delta g, \Delta H, \Delta h]_{6x1}$
STATE	Vector of relative position and velocity $[\Delta r_I, \Delta r_J, \Delta r_K, \Delta V_I, \Delta V_J, \Delta V_K]_{6x1}$
HCURL	$\mathcal{H}$ matrix for measurement $[\mathcal{H}]_t$ 2x6
DELDEL	Update vector for relative element state vector, computed by estimator
WKAREA, WK	Work matrix for inversion subroutine
ROTT	Rotation matrix PQW $\rightarrow$ IJK, 3x3
ROTI	Rotation matrix
POS	Position vector in PQW frame
VEL	Velocity vector in PQW frame
HH	Matrix of for all measurements $[\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3 \cdots \mathcal{H}_N]_{2nx6}$
HHTP	HH Transpose
QINV	Q matrix inverse
HHTPQ	$\mathcal{H}^T Q^{-1}$ used in least squares equation
HTPQH	$\mathcal{H}^T Q^{-1} \mathcal{H}$
Res	Residual vector
HTPQHI	$(\mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$
PHTP	$(\mathcal{H}^T Q^{-1} \mathcal{H})^{-1} \mathcal{H}^T$
PHTPQ	$(\mathcal{H}^T Q^{-1} \mathcal{H})^{-1} \mathcal{H}^T Q^{-1}$
POSI	Position vector in inertial frame
VELI	Velocity vector in inertial frame
POST	Position vector of target in PQW frame

VELT	Velocity vector of target in PQW frame
POSTI	Position vector of target in inertial frame
VELTI	Velocity vector of target in inertial frame
HMAT	H matrix for measurement time
RNG	Vector of range measurements
RRT	Vector of range rate measurements
POSS	Storage matrix for position vectors
VELS	Storage matrix for velocity vectors
TSTOR	Vector of time of measurement vectors
PM	Initial inverse covariance matrix computed for Bayes filter, $P^{-1}(-)$
DELM	Initial vector of differences of elements for Bayes filter, $\bar{\delta}(-)$
HCURLT	$\mathcal{H}^T$

The following matrices/vectors are used in the Bayes filter of the main program

M1	$\mathcal{H}^T Q^{-1}$
PP	$P(-) + \mathcal{H}^T Q^{-1}$
PP1	$(P(-) + \mathcal{H}^T Q^{-1} \mathcal{H})^{-1}$
M3	$\mathcal{H}^T Q^{-1} r$
DELDIF	$\bar{\delta}(-) - \bar{\delta}_{ESTIMATED}$
M4	$P(-)^{-1} (\bar{\delta}(-) - \bar{\delta}_{EST})$
TOT	$M_3 + M_4$
DELDEL	PP1 x TOT, DELDEL is the Bayes filter computed updates to the relative element state vector
CC	Vector composed of the square roots of the diagnol elements of the Bayes computed covariance matrix

RELATIVE ORBITAL ELEMENT ESTIMATOR PROGRAM

```
      PROGRAM SEOFIL
*****************************************************************

      RELATIVE ORBITAL ELEMENT ESTIMATOR PROGRAM

*****************************************************************
      REAL L,LS,G,GS,H,HS,E,MU,ECC,DYFDE,DXDDE,DYWDE,DVPDE,DVDDE,
     *DVWDE,XPPL,XDPL,XWPL,VPPL,VDPL,VWPL,DEDL,DEDLS,XPPG,XDPG,VPPG,
     *VDPG,DEDG,[illegible]X,[illegible],LNOD,[illegible]GP,INCL,TAMS,RANGE,[illegible]RATE,ANGES,RRATES,
     *ECCTGT,E[illegible],ECINT,SIGR,SIGRR,GSI,HSI,LI,GI,HI,[illegible]E[illegible]N[illegible]D,[illegible]ST,LSD,[illegible],
     *DLSCDE,CK,C1,C2,C[illegible],C[illegible],C[illegible],C[illegible],TIMEM,[illegible]KM,[illegible]KM,INT,NOSS,NOSRR,
     *DL,DL1,DLS,DLS1,DG,DG1,DGS,DGS1,DH,DH1,DHS,DHS1,[illegible]
      INTEGER B1,NUMMEA,B2,XY,ID,CYCLE,J,K,[illegible],[illegible],IER,[illegible],COUNT,[illegible],STORE,
     *NOSE,ID1,MODE,SEED
      REAL PAPG([illegible],3),PAPH([illegible],3),PRPHS(6,6),PRPGS(6,6),XPOW(6,1),R(6,6),
     *DXDL(6,1),DXDLS(6,1),DXDG([illegible],1),XIDLS(6,1),XIDL(6,1),INTER(6,1),
     *XIDG(6,1),XIDGS(6,1),XIDH(6,1),XIDHS(6,1),TRANS([illegible],[illegible]),
     *DEL([illegible],1),STATE(6,[illegible]),HOUT[illegible](7,1),DELDEL(6,1),[illegible]KA[illegible]EA(2[illegible]),ROTT(3,3),
     *ROTI(3,3),[illegible]K([illegible]),POS(3,1),VEL(3,1),HH([illegible],6),HHTP(6,[illegible]),DINV([illegible],[illegible])
     *,HHTPO(6,[illegible]),HTPOH(6,6),[illegible]([illegible],1),HTPOHI(6,6),PHIF(6,[illegible]),
     *PHTPO(6,[illegible]),POSI(3,1),VELI(3,1),POST(3,1),VELT(3,1),POSTI(3,1),
     *VELTI(3,1),HMAT(2,6),EIG(6),EIGV(6,6),RNG(2[illegible]),RRT(2[illegible]),
     *POSS(3,2[illegible]),VELS(3,2[illegible]),TSTOR(2[illegible]),PM(6,6),DELM(6,1),MOURLT(6,2),
     *M2(6,6),PP(6,[illegible]),PP1([illegible],[illegible]),KT(3,1),M1([illegible],[illegible]),
     *DELDIF(6,1),TOT(6,1),[illegible]([illegible],1),CD([illegible])
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
      COUNT=
      MU=1.
      STORE=
      CYCLE=
      D1 =
      PI=3.1415926[illegible]
      SIGR=1.[illegible]E-[illegible]
      SIGRR=3.[illegible]E-[illegible]
      PRINT *,' INPUT MODE OF FILTER,1 IF LEASTT SSQUARES,[illegible] IF BAYES '
      READ *,MODE
      STATE VECTOR [illegible] IS DEFINED AS THE VECTOR OF RELATIVE DELAUNAY ELEMENTS
      PRINT *,' INPUT NOISE PARAMETER,IF =1 THEN NOISE IS ADDED TO MEASU
     *REMENTS '
      READ *,NOSE
      PRINT *,' INPUT RANDOM NUMBER GENERATOR SEED '
      READ *,SEED
      CALL RANSET(SEED)
      PRINT *,' ENTER NUMBER OF INITIAL MEASUREMENTS TO BE TAKEN '
      READ *,NUMMEA
      PRINT *,' NUMMEA= ',NUMMEA
      B1=NUMMEA*[illegible]
      DO [illegible] I=1,[illegible]
        DO 4 J=1,B1
          DINV(I,J)= .
4     CONTINUE
5     CONTINUE
      B2=B1-1
      DO 1[illegible] I=1,B2,[illegible]
        DINV(I,I)=[illegible].[illegible]E+[illegible]
10    CONTINUE
      DO 2[illegible] I=2,B1,[illegible]
      DINV(I,I)=[illegible].[illegible]E+[illegible]
20    CONTINUE
      PRINT *,' ENTER SEMI MAJOR AXIS(DU),ECCENTRICITY,LONGITUDE OF '
      PRINT [illegible],'[illegible] ASCENDING NODE,ARG OF PERIGEE,INCLINATION(DEGREES) '
```

```
READ *,AX,ECC,INCL,ARGP,NODE
PRINT *,' ELEMENTS ',AX,ECC,INCL,ARGP,INCL
GS=ARGP*PI/180.
HS=NODE*PI/180.
L=SQRT(AX)
G=L*((1.-ECC**2)**.5)
INCL=INCL*PI/180.
H=G*COS(INCL)
CALL ROT33(ROTI)
GSI=GS
HSI=HS
LI=L
GI=G
HI=H
XX=-1
PRINT *,' INPUT TRUTH MODEL DIFFERENCES BETWEEN TARGET AND '
PRINT *,' INTERCEPTOR ELEMENTS '
READ *,DL,DLS,DG,DGS,DH,DHS
PRINT *,' TRUTH MODEL DELTAS '
PRINT *,DL,' ',DLS,' ',DG,' ',DGS
PRINT *,DH,' ',DHS
PRINT *,' ENTER PERTURBATION TO TRUTH MODEL DELTA ELEMENTS '
READ *,PTB
PRINT *,' PTB = ',PTB
RANDOM PERTURBATION TO TRUTH MODEL

CK=RANF()
IF (CK .LE. 0.5) THEN
DEL(1,1)=DL+PTB*DL
ELSE
DEL(1,1)=DL-PTB*DL
ENDIF
CK=RANF()
IF (CK .LE. .5) THEN
DEL(2,1)=DLS+PTB*DLS
ELSE
DEL(2,1)=DLS-PTB*DLS
ENDIF
CK=RANF()
IF (CK .LE. .5) THEN
DEL(3,1)=DG+PTB*DG
ELSE
DEL(3,1)=DG-PTB*DG
ENDIF
CK=RANF()
IF (CK .LE. .5) THEN
DEL(4,1)=DGS+PTB*DGS
ELSE
DEL(4,1)=DGS-PTB*DGS
ENDIF
CK=RANF()
IF (CK .LE. .5) THEN
DEL(5,1)=DH+PTB*DH
ELSE
DEL(5,1)=DH-PTB*DH
ENDIF
CK=RANF()
IF (CK .LE. .5) THEN
DEL(6,1)=DHS+PTB*DHS
ELSE
DEL(6,1)=DHS-PTB*DHS
ENDIF
PRINT *,' INITIAL ESTIMATE OF THE RELATIVE ELEMENT VECTOR '
PRINT *,' ELEMENTS(ESTIMATED) - INTERCEPTOR '
PRINT *,' DEL',DEL(1,1),DEL(2,1),DEL(3,1),DEL(4,1),DEL(5,1),
*DEL(6,1)
```

```
30    CONTINUE
      COUNT=COUNT+1
***       RADAR MEASUREMENT        **
      PRINT *,' ENTER RANGE, RNGE RATE,TIME,ID       '
      PRINT *,' ID=  IF MEASUREMENT IS LAST OF INITIAL SET '
      READ *,RANGE,RRATE,TIME,ID
**************        NOISE INPUT OPTION        ***************
      IF (NOSE .EQ. 1) THEN
      NOSR=(RANF()-.5)*2*1.732 1.5 -5
      NOSRR=(RANF()-.5)*2*1.732*3.5 -5
        CALL ADDN(NOSR,NOSRR,RANGE,RRATE)
      ENDIF
      PRINT *,' INITIAL MEASUREMENT  ',COUNT,'  DATA  '
      PRINT *,' '
      PRINT *,' RANGE = ',RANGE,' RRATE = ',RRATE,' TIME= ',TIME
      PRINT *,' '
      RKM=RANGE*6378.145
      RRKM=RRATE*7.9053613 2
      TIMEM=TIME*13.44689
      PRINT *,' RANGE(KM)= ',RKM,'  RRATE(KM/SEC)= ',RRKM,'  TIME(MIN)
     *= ',TIMEM
      STORE=STORE+1
      RNG(STORE)=RANGE
      RRT(STORE)=RRATE
      TSTOR(STORE)=TIME
***       COMPUTE INTERCEPTOR POSITION AND VELOCITY      **
      L=LI
      CALL KEPLER(TIME,ECC,E)
      LSI=E-ECC*SIN(E)
      CALL POSVEL(LI,GI,ECC,E,POS,VEL)
      CALL MULTI(ROTI,POS,POSI,3)
      CALL MULTI(ROTI,VEL,VELI,3)
      DO 31 I=1,3
        POSS(I,STORE)=POSI(I,1)
        VELS(I,STORE)=VELI(I,1)
31    CONTINUE
40    CONTINUE
***       COMPUTE TARGET POSITION AND VELOCITY      ***
      IF (ID .EQ. 1) THEN
      STORE=STORE+1
      ENDIF
      C1=C1+1
      XX=XX+Z
      L=LI+DEL(1,1)
      G=GI+DEL(3,1)
      GS=GSI+DEL(4,1)
      H=HI+DEL(5,1)
      HS=HSI+DEL(6,1)
      ECCTGT=SQRT(1.-G**2/L**2)
      MEANMO=(1./L**3)
      LST=MEANMO*TSTOR(STORE)+DEL(2,1)
      CALL ROTEE(ROTT)
      ECENT=LST
50    LSC=ECENT-ECCTGT*SIN(ECENT)
      DLSCDE=1.-ECCTGT*COS(ECENT)
      ECENT=ECENT+(LST-LSC)/DLSCDE
      CK=LST-LSC
      CK=ABS(CK)
      IF (CK .LT. 1E-12) GO TO 60
      GO TO 50
60    CONTINUE
      E=ECENT
      LS=LST
      CALL PHI(TRANS)
      CALL POSVEL(L,G,ECCTGT,E,POST,VELT)
      CALL MULTI(ROTT,POST,POSTI,3)
```

```
      CALL MULT1(ROT?,VELT,VEL??.?)
C***    COMPUTE RELATIVE POSITION AND VELOCITY     ***
      DO 70 I=1,3
        STATE(I,1)=POSTI(I,1)-POS(I,STORE)
        STATE(I+3,1)=VELTI(I,1)-VELS(I,STORE)
 70   CONTINUE
C***    NON LINEAR LEAST SQUARES ESTIMATION     ***
      CALL HM(STATE,HMAT)
      CALL MULT1(HMAT,TRANS,HOUEL,2,6,6)
      RANGEG=SQRT(STATE(1,1)**2+STATE(2,1)**2+STATE(3,1)**2)
      RRATEG=(STATE(4,1)*STATE(1,1)+STATE(5,1)*STATE(2,1)+
     *STATE(6,1)*STATE(3,1))/RANGEG
      DO 81 I=1,6
        HH(XX,I)=HOUEL(1,I)
 81   CONTINUE
      DO 82 I=1,6
        HH(XX+1,I)=HOUEL(2,I)
 82   CONTINUE
C***    COMPUTE RANGE AND RANGE RATE RESIDUALS     ***
      RES(XX,1)=RNG(STORE)-RANGEG
      RES(XX+1,1)=RRT(STORE)-RRATEG
      IF (ID .EQ. 1) GO TO 3?
      IF (C1  .EQ. NUMBEA) GO TO 88
      GO TO ?
 88   CONTINUE
      ID=?
      RMSR=
      RMSRR=
      DO 90 I=1,?2,2
        RMSR=RMSR+RES(I,1)**2
        RMSRR=RMSRR+RES(I+1,1)**2
 90   CONTINUE
      RMSR=SQRT(RMSR/NUMBEA)
      RMSRR=SQRT(RMSRR/NUMBEA)
      PRINT *,' RMS RANGE AND RANGE RATE RESIDUAL ERRORS BEFORE FILTER '
      PRINT *,' RMSR = ',RMSR,' RMSRR =   ',RMSRR
      CYCLE=CYCLE+1
C***    NON LINEAR LEAST SQUARES ESTIMATION EQUATION     ***
      CALL TRPOS(HH,HHTR,B1,6,..)
      CALL MULT1(HHTR,TINV,HHTRT,?1,6,B1,.,.)
      CALL MULT1(HHTRT,HH,HTRH,?,.,.,.,.)
      CALL LINV2F(HTRH,.,.,HTRHI,.,WKAREA,IER)
      CALL MULT1(HTRHI,HHTR,PHTR,.,.,?1,.,.)
      CALL MULT1(PHTR,TINV,PHTRT,?,.,B1,.,4?)
      CALL MULT1(PHTRT,RES,DELDEL,?,.,1,.,.)
      PRINT *,' THE CHANGES TO THE STATE VECTOR COMPUTED BY THE FILTER '
      CALL MPRINT(DELDEL,6,1,6)

C***    UPDATE RELATIVE ORBITAL ELEMENT STATE VECTOR     ***

      DO 91 I=1,6
        DEL(I,1)=DEL(I,1)+DELDEL(I,1)
        DELDEL(I,1)=ABS(DELDEL(I,1))
 91   CONTINUE
      DO 93 I=1,?
        DO 92 J=1,6
          PM(I,J)=HTRHI(I,J)
 92     CONTINUE
 93   CONTINUE
C***    UPDATE TARGET ELEMENTS     ***
      L=LI+DEL(1,1)
      G=GI+DEL(3,1)
      GS=GSI+DEL(4,1)
      H=HI+DEL(5,1)
      HS=HSI+DEL(6,1)
      ECCT=SQRT(?,+G??2/ ??2)
```

```
      MEANMO=SQRT(1./(L**3))
      CALL ROT3F(ROTT)
      XX=-1
      DO 97 I=1,NUMMEA
      XX=XX+2
      ***   RECOMPUTE TARGET POSITION AND VELOCITY BASED UPON    ***
      ***   UPDATED ELEMENTS                                     ***
      LST=MEANMO*TSTOR(I)+DEL(3,1)
      ECENT=LST
 94   LSC=ECENT-ECCTGT*SIN(ECENT)
      DLSCDE=1.-ECCTGT*COS(ECENT)
      ECENT=ECENT+(LST-LSC)/DLSCDE
      CK=LST-LSC
      CK=ABS(CK)
      IF (CK .LT. 1.E-10) GO TO 95
      GO TO 94
 95   CONTINUE
      E=ECENT
      CALL POSVEL(L,G,ECCTGT,E,POST,VELT)
      CALL MULTI(ROTT,POST,POSTI,3)
      CALL MULTI(ROTT,VELT,VELTI,3)
      ***   RECOMPUTE RELATIVE POSITION AND VELOCITY STATE VECTOR   ***
      DO 96 J=1,3
        STATE(J,1)=POSTI(J,1)-POBS(J,I)
        STATE(J+3,1)=VELTI(J,1)-VOBS(J,1)
 96   CONTINUE
      RANGEG=SQRT(STATE(1,1)**2+STATE(2,1)**2+STATE(3,1)**2)
      RRATEG=(STATE(4,1)*STATE(1,1)+STATE(5,1)*STATE(2,1)+
     *STATE(6,1)*STATE(3,1))/RANGEG
      ***   COMPUTE RANGE AND RANGE RATE RESIDUALS   ***
      RES(XX,1)=RNG(I)-RANGEG
      RES(XX+1,1)=RRT(I)-RRATEG
 97   CONTINUE
      RMSR=
      RMSRR=
      DO 98 I=1,32,2
        RMSR=RES(I,1)**2+RMSR
        RMSRR=RES(I+1,1)**2+RMSRR
 98   CONTINUE
      RMSR=SQRT(RMSR/NUMMEA)
      RMSRR=SQRT(RMSRR/NUMMEA)
      PRINT *,' RMS RANGE AND RANGE RATE RESIDUAL ERRORS AFTER  FILTER '
      PRINT *,'  RMSR = ',RMSR,' RMSRR = ',RMSRR
      DO 99 I=1,31
        RES(I,1)=ABS(RES(I,1))
 99   CONTINUE
      DC(1)=SQRT(HTRPHI(1,1))
      DC(2)=SQRT(HTRPHI(2,2))
      DC(3)=SQRT(HTRPHI(3,3))
      DC(4)=SQRT(HTRPHI(4,4))
      DC(5)=SQRT(HTRPHI(5,5))
      DC(6)=SQRT(HTRPHI(6,6))
      ***   CONVERGENCE  CHECK   ***
      COUNT=
      DO 1    I=1,6
      IF (DELDEL(I,1) .LE. DC(I)) THEN
      COUNT=COUNT+1
       ENDIF
 100  CONTINUE
      PRINT *,' COUNT = ',COUNT,' FOR DELTA CHECK '
      CNT=
      DO 12  I=1,32,2
      IF (RES(I,1) .LT. 3*SIGR) THEN
      CNT=CNT+1
      ENDIF
 12   CONTINUE
```

```
      ***   COMPUTE RANGE AND RANGE RATE RESIDUALS   ***
      RES(YX,1)=RNG(STORE)-RANGER
      RES(YX+1,1)=RRT(STORE)-RRATER
      IF (ID .EQ. 1) GO TO 3?
      IF (C1? .EQ. NUMNEA) GO TO 38
      GO TO 320
 380  CONTINUE
      RMSR=0
      RMSRR=
      DO 382 I=1,32,2
        RMSR=RES(I,1)**2+RMSR
        RMSRR=RES(I+1,1)**2+RMSRR
 382  CONTINUE
      RMSR=SQRT(RMSR/NUMMEA)
      RMSRR=SQRT(RMSRR/NUMMEA)
      PRINT *,' RMS RANGE AND RANGE RATE RESIDUAL ERRORS BEFORE FILTER '
      PRINT *,' RMSR = ',RMSR,' RMSRR = ',RMSRR
      ID1=7
      CYCLE=CYCLE+1
      PRINT *,' CYCLE = ',CYCLE
      ***   BAYES ESTIMATION EQUATION    ***
      CALL TRPOS(HH,HHTP,B2,6, ,6)
      CALL MULT1A(HHTP,QINV,HHTPQ,?1,6,?1,6,  )
      CALL MULT1A(HHTPQ,HH,HTPQH,?1,6,6,6,  )
      CALL MULT1A(HHTP,QINV,M1,P1,6,?1,?,?  )
      DO 391 I=1,6
        DO 39? J=1,6
          PP(I,J)=PM(I,J)+HTPQH(I,J)
 390  CONTINUE
 391  CONTINUE
      CALL LINV2F(PP,6,6,PP1, ,WKAREA,IER)
      CALL MULT1A(M1,RES,M3,?1,6,1,6,  )
      DO 4   I=1,6
        DELDIF(I,1)=DELM(I,1)-DEL(I,1)
 400  CONTINUE
      CALL MULT1(PM,DELDIF,M ,6,6,1)
      DO 41  I=1,6
        TOT(I,1)=M3(I,1)+M (I,1)
 410  CONTINUE
      CALL MULT1(PP1,TOT,DELDEL,6,6,1)
      PRINT *,' THE CHANGES TO THE STATE VECTOR COMPUTED BY THE FILTER '
      CALL MPRINT(DELDEL,6,1,6)
      ***   UPDATE RELATIVE ORBITAL ELEMENT STATE VECTOR   ***
      DO 420 I=1,6
      DEL(I,1)=DEL(I,1)+DELDEL(I,1)
      DELDEL(I,1)=ABS(DELDEL(I,1))
 420  CONTINUE
      DC(1)=SQRT(PP1(1,1))
      DC(2)=SQRT(PP1(2,2))
      DC(3)=SQRT(PP1(3,3))
      DC( )=SQRT(PP1(4,4))
      DC(5)=SQRT(PP1(5,5))
      DC( )=SQRT(PP1(6,6))
      DO 430 I=1,6
      IF (DELDEL(I,1) .LE. DC(I)) THEN
      COUNT=COUNT+1
      ENDIF
 430  CONTINUE
      PRINT *,' COUNT = ',COUNT,' FOR DELTA CHECK '
      CNT=
      ***   UPDATE TARGET ELEMENTS   ***
      L=LI+DEL(1,1)
      G=GI+DEL(3,1)
      GS=GSI+DEL(4,1)
      H=HI+DEL(?,1)
      HS=HSI+DEL(5,1)
```

```
      ECCTGT=SQRT(1.-G**2/L**2)
      MEANMO=SQRT(1./(L**3))
C***     RECOMPUTE TARGET POSITION AND VELOCITY BASED UPON      ***
C***     UPDATED ELEMENTS                                       ***
      CALL ROT33(ROTT)
      XX=-1
      DO 437 I=1,NUMMEA
      XX=XX+2
      LST=MEANMO*TSTOR(I)+DEL(3,1)
      ECENT=LST
434   LSC=ECENT-ECCTGT*SIN(ECENT)
      DLSCDE=1.-ECCTGT*COS(ECENT)
      ECENT=ECENT+(LST-LSC)/DLSCDE
      CK=LST-LSC
      CK=ABS(CK)
      IF (CK .LT. 1E-10) GO TO 435
      GO TO 434
435   CONTINUE
      E=ECENT
      CALL POSVEL(L,G,ECCTGT,E,POST,VELT)
      CALL MULTI(ROTT,POST,POSTI,3)
      CALL MULTI(ROTT,VELT,VELTI,3)
C***     RECOMPUTE RELATIVE POSITION AND VELOCITY     ***
      DO 436 J=1,3
        STATE(J,1)=POSTI(J,1)-POSS(J,1)
        STATE(J+3,1)=VELTI(J,1)-VELS(J,1)
436   CONTINUE
      RANGEG=SQRT(STATE(1,1)**2+STATE(2,1)**2+STATE(3,1)**2)
      RRATEG=(STATE(4,1)*STATE(1,1)+STATE(5,1)*STATE(2,1)+
     *STATE(6,1)*STATE(3,1))/RANGEG
C***     COMPUTE RANGE AND RANGE RATE RESIDUALS     ***
      RES(XX,1)=RNG(I)-RANGEG
      RES(XX+1,1)=RRT(I)-RRATEG
437   CONTINUE
      RMSR=0
      RMSRR=0
      DO 438 I=1,32,2
        RMSR=RES(I,1)**2+RMSR
        RMSRR=RES(I+1,1)**2+RMSRR
438   CONTINUE
      RMSR=SQRT(RMSR/NUMMEA)
      RMSRR=SQRT(RMSRR/NUMMEA)
      PRINT *,' RMS RANGE AND RANGE RATE RESIDUAL ERRORS AFTER  FILTER '
      PRINT *,' RMSR = ',RMSR,' RMSRR =  ',RMSRR
C***     CONVERGENCE  CHECK     ***
      DO 440 I=1,31
        RES(I,1)=ABS(RES(I,1))
440   CONTINUE
      DO 450 I=1,32,2
      IF (RES(I,1) .LT. 3*SIGR) THEN
          CNT=CNT+1
        ENDIF
450   CONTINUE
      DO 460 I=2,31,2
      IF (RES(I,1) .LT. 3*SIGRR) THEN
        CNT=CNT+1
      ENDIF
460   CONTINUE
      PRINT *,' COUNT = ',CNT,' FOR RESIDUAL CHECK  '
      IF (CNT .EQ. 31) THEN
        GO TO 500
      ENDIF
      IF (CYCLE .EQ. 6) THEN
        PRINT *,' PROGRAM ABORT IN SEQUENTIAL MODE '
        GO TO 999
      ENDIF
```

```
      COUNT=
      C1 =
      XX=-1
      STORE=.
      GO TO 32
500   CONTINUE
***   OUTPUT FINAL RELATIVE ORBITAL ELEMENT STATE VECTOR   ***
      PRINT *,'  RELATIVE ELEMENT STATE VECTOR IS '
      CALL MPRINT(DEL,6,1,6)
      PRINT *,' COVARIANCE MATRIX IS '
      CALL MPRINT(PP1,6,6,6)
      IF (ID .EQ. 999) GO TO 999
      CALL LINV2F(PP1,6,6,PM,,WKAREA,IER)
      DO 510 I=1,6
      DELM(I,1)=DEL(I,1)
510   CONTINUE
      COUNT=
      CNT=
      STORE=
      CYCLE=
      C1=.
      XX=-1
      GO TO 3
999   CONTINUE
      PRINT *,'  RADAR DATA AND RUN COMPLETE '
      END
```

Appendix E

Subroutines Used in Computer Programs

```
      SUBROUTINE RVPOW(XPOW)
      REAL  L,LS,G,GS,H,HS,E,ECC,MU
      REAL XPOW(1:6,1:1)
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
      XPOW(1,1)=((L**2)/MU)*(COS(E)-ECC)
      XPOW(2,1)=(L*G/MU)*(SIN(E))
      XPOW(3,1)=
      XPOW(4,1)=(-MU)*SIN(E)/(L*(1.-ECC*COS(E)))
      XPOW(5,1)=(MU)*G*COS(E)/(L**2*(1.-ECC*COS(E)))
      XPOW(6,1)=
      END


      ROTATE COMPUTES A 6X6 ROTATION MATRIX FOR THE POSITION VELOCITY
      VECTOR FROM THE POW TO THE IJK FRAME

      SUBROUTINE ROTATE(R)
      REAL  L,LS,G,GS,H,HS,E,ECC,MU
      INTEGER I,J
      REAL R(1:6,1:6)
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
      R(1,1)=COS(HS)*COS(GS)-SIN(HS)*SIN(GS)*H/G
      R(1,2)=(-1.0)*(COS(HS)*SIN(GS)+SIN(HS)*COS(GS)*H/G)
      R(1,3)=SIN(HS)*SQRT(1.0-H**2/G**2)
      R(2,1)=SIN(HS)*COS(GS)+COS(HS)*SIN(GS)*H/G
      R(2,2)=-SIN(HS)*SIN(GS)+COS(HS)*COS(GS)*H/G
      R(2,3)=-COS(HS)*SQRT(1.0-H**2/G**2)
      R(3,1)=SIN(GS)*SQRT(1.0-H**2/G**2)
      R(3,2)= COS(GS)*SQRT(1.0-H**2/G**2)
      R(3,3)=H/G
      R(6,6)=R(3,3)
      R(6,5)=R(3,2)
      R(6,4)=R(3,1)
      R(5,6)=R(2,3)
      R(5,5)=R(2,2)
      R(5,4)=R(2,1)
      R(4,6)=R(1,3)
      R(4,5)=R(1,2)
      R(4,4)=R(1,1)
      DO 20  I=1,3
      DO 10  J=4,6
        R(I,J)=0
 10   CONTINUE
 20   CONTINUE
      DO 40  I=4,6
      DO 30  J=1,3
        R(I,J)=0
 30   CONTINUE
 40   CONTINUE
      END


      PART COMPUTES THE PARTIAL DERIVATIVE MATRICES OF THE ROTATION
      MATRIX WITH RESPECT TO THE DELAUNAY ELEMENTS G,GS,H,HS

      SUBROUTINE PART(PRPG,PRPGS,PRPH,PRPHS)


      THIS SUBROUTINE COMPUTES FOUR MATRICES OF PARTIALS OF THE
      ROTATION MATRIX WITH RESPECT TO G,GS,H,HS.


      REAL  L,LS,G,GS,H,HS,E,ECC,MU
```

```
      INTEGER I,J
      REAL PRPG(1*6,1*6),PRPGS(1*6,1*6),PRPH(1*6,1*6),PRPHS(1*6,1*6)
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU

      PRPG

      PRPG(1,1)=SIN(HS)*SIN(GS)*H/G**2
      PRPG(1,2)=SIN(HS)*COS(GS)*H/G**2
      PRPG(1,3)=(SIN(HS)*(H**2/G**3))/SQRT(1.0-H**2/G**2)
      PRPG(2,1)=-COS(HS)*SIN(GS)*H/G**2
      PRPG(2,2)=-COS(HS)*COS(GS)*H/G**2
      PRPG(2,3)=(-COS(HS)*(H**2/G**3))/SQRT(1.0-H**2/G**2)
      PRPG(3,1)=(SIN(GS)*(H**2/G**3))/SQRT(1.0-H**2/G**2)
      PRPG(3,2)=(COS(GS)*(H**2/G**3))/SQRT(1.0-H**2/G**2)
      PRPG(3,3)=-H/G**2
      DO 10 I=1,3
      DO 5 J=4,6
        PRPG(I,J)=0.
5     CONTINUE
10    CONTINUE
      DO 20 I=4,6
      DO 15 J=1,3
        PRPG(I,J)=0.
15    CONTINUE
20    CONTINUE
      PRPG(4,4)=PRPG(1,1)
      PRPG(5,4)=PRPG(2,1)
      PRPG(6,4)=PRPG(3,1)
      PRPG(4,5)=PRPG(1,2)
      PRPG(5,5)=PRPG(2,2)
      PRPG(6,5)=PRPG(3,2)
      PRPG(4,6)=PRPG(1,3)
      PRPG(5,6)=PRPG(2,3)
      PRPG(6,6)=PRPG(3,3)

      PRPH

      PRPH(1,1)=(-SIN(HS)*SIN(GS)/G)
      PRPH(1,2)=(-SIN(HS)*COS(GS)/G)
      PRPH(1,3)=(-SIN(HS)*H/G**2)/SQRT(1.0-H**2/G**2)
      PRPH(2,1)=COS(HS)*SIN(GS)/G
      PRPH(2,2)=COS(HS)*COS(GS)/G
      PRPH(2,3)=(COS(HS)*H/G**2)/SQRT(1.0-H**2/G**2)
      PRPH(3,1)=(-SIN(GS)*H/G**2)/SQRT(1.0-H**2/G**2)
      PRPH(3,2)=(-COS(GS)*H/G**2)/SQRT(1.0-H**2/G**2)
      PRPH(3,3)=1. /G
      DO 25 I=1,3
      DO 21 J=4,6
        PRPH(I,J)=0.
21    CONTINUE
25    CONTINUE
      DO 35 I=4,6
      DO 30 J=1,3
        PRPH(I,J)=0.
30    CONTINUE
35    CONTINUE
      PRPH(4,4)=PRPH(1,1)
      PRPH(5,4)=PRPH(2,1)
      PRPH(6,4)=PRPH(3,1)
      PRPH(4,5)=PRPH(1,2)
      PRPH(5,5)=PRPH(2,2)
      PRPH(6,5)=PRPH(3,2)
      PRPH(4,6)=PRPH(1,3)
      PRPH(5,6)=PRPH(2,3)
      PRPH(6,6)=PRPH(3,3)
```

```
      PRPHS

      PRPHS(1,1)=-(SIN(HS)*COS(GS)+(H/G)*SIN(GS)*COS(HS))
      PRPHS(1,2)=SIN(HS)*SIN(GS)-COS(HS)*COS(GS)*H/G
      PRPHS(1,3)=COS(HS)*SQRT(1.-H**2/G**2)
      PRPHS(2,1)=COS(HS)*COS(GS)-SIN(HS)*SIN(GS)*H/G
      PRPHS(2,2)=-(COS(HS)*SIN(GS)+SIN(HS)*COS(GS)*H/G)
      PRPHS(2,3)=SIN(HS)*SQRT(1.-H**2/G**2)
      DO 45 I=1,3
      DO 40 J=4,6
        PRPHS(I,J)=
 40   CONTINUE
 45   CONTINUE
      DO 55 I=4,6
      DO 50 J=1,3
        PRPHS(I,J)=
 50   CONTINUE
 55   CONTINUE
      PRPHS(3,1)=0
      PRPHS(3,2)=0
      PRPHS(3,3)=0
      PRPHS(4,4)=PRPHS(1,1)
      PRPHS(5,4)=PRPHS(2,1)
      PRPHS(6,4)=PRPHS(3,1)
      PRPHS(4,5)=PRPHS(1,2)
      PRPHS(5,5)=PRPHS(2,2)
      PRPHS(6,5)=PRPHS(3,2)
      PRPHS(4,6)=PRPHS(1,3)
      PRPHS(5,6)=PRPHS(2,3)
      PRPHS(6,6)=PRPHS(3,3)

      PRPGS

      PRPGS(1,1)=-(COS(HS)*SIN(GS)+SIN(HS)*COS(GS)*H/G)
      PRPGS(1,2)=SIN(HS)*SIN(GS)*H/G-COS(HS)*COS(GS)
      PRPGS(1,3)=0
      PRPGS(2,1)=-SIN(HS)*SIN(GS)+COS(HS)*COS(GS)*H/G
      PRPGS(2,2)=-(SIN(HS)*COS(GS)+COS(HS)*SIN(GS)*H/G)
      PRPGS(2,3)=0
      PRPGS(3,1)=COS(GS)*SQRT(1.0-H**2/G**2)
      PRPGS(3,2)=-SIN(GS)*SQRT(1.0-H**2/G**2)
      PRPGS(3,3)=0
      DO 65 I=1,3
      DO 60 J=4,6
        PRPGS(I,J)=
 60   CONTINUE
 65   CONTINUE
      DO 75 I=4,6
      DO 70 J=1,3
        PRPGS(I,J)=
 70   CONTINUE
 75   CONTINUE
      PRPGS(4,4)=PRPGS(1,1)
      PRPGS(5,4)=PRPGS(2,1)
      PRPGS(6,4)=PRPGS(3,1)
      PRPGS(4,5)=PRPGS(1,2)
      PRPGS(5,5)=PRPGS(2,2)
      PRPGS(6,5)=PRPGS(3,2)
      PRPGS(4,6)=PRPGS(1,3)
      PRPGS(5,6)=PRPGS(2,3)
      PRPGS(6,6)=PRPGS(3,3)
      END
```

XDS COMPUTES THE DERIVATIVE OF THE POSITION AND VELOCITY VECTOR

AD-A111 107 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 22/3
DESIGN OF AN ORBITAL ELEMENT ESTIMATOR USING RELATIVE MOTION DA--ETC(U)
DEC 81 J F ANTHONY
UNCLASSIFIED AFIT/GA/AA/81D-1 NL

2 OF 2
AD
A111107

END
DATE
FILMED
3-82
DTIC

```
 WITH RESPECT TO THE ECCENTRIC ANOMALY

 SUBROUTINE XDE(DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE)
 REAL ECC,MU,L,E,G,DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,LS
 COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
 DXPDE=-(L**2/MU)*SIN(E)
 DXQDE=L*G*COS(E)/MU
 DXWDE=0
 DVPDE=(MU*(SIN(E)**2)*L*ECC-MU*COS(E)*L*(1.0-ECC*COS(E)))/
*((L*(1.0-ECC*COS(E)))**2)
 DVQDE=-((L*L*(1.0-ECC*COS(E)))*(MU*G*SIN(E))+
*(MU*G*COS(E)*SIN(E)*L*L*ECC))/((L*L*(1.0-ECC*COS(E)))**2)
 DVWDE=0
 END


 XDL COMPUTES THE DERIVATIVE OF THE POSITION AND VELOCITY VECTOR
 WITH RESPECT TO THE DELAUNAY ELEMENT L
 SUBROUTINE XDL(DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,DXDL)
 REAL DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,ECC,MU,XPPL,XQPL,
*XWPL,VPPL,VQPL,VWPL,DEDL,L,LS
 REAL DXDL(1:6,1:1)
 COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
 DEDL=((1/ECC)-G*G*SIN(E)/L**3)/(1.0-ECC*COS(E))
 XPPL=2*L*COS(E)/MU-2*L*ECC/MU-L*L/ECC/MU*G*G/L**3
 XQPL=G*SIN(E)/MU
 XWPL=0
 VPPL=MU*SIN(E)*(1.0-ECC*COS(E)-L/ECC*G*G*COS(E)/L**3)/
*(L*(1.0-ECC*COS(E)))**2
 VQPL=-MU*G*COS(E)*(2*L*(1.0-ECC*COS(E))-L*L/ECC*G*G*COS(E)/
*L**3)/(L*L*(1.0-ECC*COS(E)))**2
 VWPL=0
 DXDL(1,1)=DXPDE*DEDL+XPPL
 DXDL(2,1)=DXQDE*DEDL+XQPL
 DXDL(3,1)=0
 DXDL(4,1)=DVPDE*DEDL+VPPL
 DXDL(5,1)=DVQDE*DEDL+VQPL
 DXDL(6,1)=0
 END


 XDLS COMPUTES THE DERIVATIVE OF THE POSITION AND VELOCITY VECTOR
 WITH RESPECT TO THE DELAUNAY ELEMENT LS

 SUBROUTINE XDLS(DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,DXDLS)
 REAL DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,ECC,MU,DEDLS,L,G,E,LS
 REAL DXDLS(1:6,1:1)
 COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
 DEDLS=1/(1.0-ECC*COS(E))
 DXDLS(1,1)=DXPDE*DEDLS
 DXDLS(2,1)=DXQDE*DEDLS
 DXDLS(3,1)=0
 DXDLS(4,1)=DVPDE*DEDLS
 DXDLS(5,1)=DVQDE*DEDLS
 DXDLS(6,1)=0
 END


 XDG COMPUTES THE DERIVATIVE OF THE POSITION AND VELOCITY VECTOR
 WITH RESPECT TO THE DELAUNAY ELEMENT G

 SUBROUTINE XDG(DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,DXDG)
 REAL XPPG,XQPG,VPPG,VQPG,MU,ECC,E,L,G,LS,DEDG,DXPDE,DXQDE,
```

```
*DXNLE,DVPDE,DVQDE,DVNDE
 REAL DXDG(1:6,1:1)
 COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
 DEDG=(1./ECC*G*SIN(E)*(L L))/(1.-ECC*COS(E))
 XPPG=G/(MU*ECC)
 XQPG=L*SIN(E)/MU
 VPPG=(MU*SIN(E)*COS(E)*G/L/ECC)/((L*(1.-ECC*COS(E)))**2)
 VQPG=(L*L*(1.-ECC*COS(E))-MU*COS(E)-MU*COS(E)*COS(E)*G*G/ECC)/
*((L*L*(1.(-ECC*COS(E)))**2)
 DXDG(1,1)=DXPDE*DEDG+XPPG
 DXDG(2,1)=DXQDE*DEDG+XQPG
 DXDG(3,1)=
 DXDG(4,1)=DVPDE*DEDG+VPPG
 DXDG(5,1)=DVQDE*DEDG+VQPG
 DXDG(6,1)=
 END


 IDL  COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
 VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT L

 SUBROUTINE IDL(R,DXDL,XIDL)
 REAL R(1:6,1:1),DXDL(1:6,1:1),XIDL(1:6,1:1)
 CALL MULTI(R,DXDL,XIDL,6)
 END


 IDLS COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
 VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT LS

 SUBROUTINE IDLS(R,DXDLS,XIDLS)
 REAL R(1:6,1:6),DXDLS(1:6,1:1),XIDLS(1:6,1:1)
 CALL MULTI(R,DXDLS,XIDLS,6)
 END

 IDG  COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
 VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT G

 SUBROUTINE IDG(PRPG,XPQW,R,DXDG,XIDG)
 INTEGER I
 REAL PRPG(1:6,1:6),R(1:6,1:6),XPQW(1:6,1:1),DXDG(1:6,1:1),
*XIDG(1:6,1:1),INTER(1:6,1:1)
 CALL MULTI(PRPG,XPQW,INTER,6)
 CALL MULTI(R,DXDG,XIDG,6)
 DO 10 I=1,6
   XIDG(I,1)=XIDG(I,1)+INTER(I,1)
10 CONTINUE
 END


 IDGS COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
 VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT GS

 SUBROUTINE IDGS(PRPGS,XPQW,XIDGS)
 REAL PRPGS(1:6,1:6),XPQW(1:6,1:1),XIDGS(1:6,1:1)
 CALL MULTI(PRPGS,XPQW,XIDGS,6)
 END


 IDH  COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
 VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT H
```

```
      SUBROUTINE IDH(PRPH,XPQW,XIDH)
      REAL PRPH(1:6,1:6),XPQW(1:6,1:1),XIDH(1:6,1:1)
      CALL MULTI(PRPH,XPQW,XIDH,6)
      END


      IDHS COMPUTES THE DERIVATIVE OF THE INERTIAL POSITION AND VELOCITY
      VECTOR WITH RESPECT TO THE DELAUNAY ELEMENT HS

      SUBROUTINE IDHS(PRPHS,XPQW,XIDHS)
      REAL PRPHS(1:6,1:6),XPQW(1:6,1:1),XIDHS(1:6,1:1)
      CALL MULTI(PRPHS,XPQW,XIDHS,6)
      END


      INERT ROTATES  POSITION AND VELOCITY VECTOR FROM PQW TO IJK FRAME

      SUBROUTINE  INERT(R,XPQW,XIN)
      REAL R(1:6,1:6),XPQW(1:6,1:1),XIN(1:6,1:1)
      CALL MULTI(R,XPQW,XIN,6)
      END


      FILL FORMS THE STATE TRANSITION/COORDINATE TRANSFORMATION MATRIX
      USING THE 6 DERIVATIVE VECTORS OF THE INERTIAL POSITION AND
      VELOCITY  WITH RESPECT TO THE DELAUNAY ELEMENTS

      SUBROUTINE FILL(XIDL,XIDLS,XIDG,XIDGS,XIDH,XIDHS,TRANS)
      REAL XIDL(1:6,1:1),XIDLS(1:6,1:1),XIDG(1:6,1:1),XIDGS(1:6,1:1),
     *XIDH(1:6,1:1),XIDHS(1:6,1:1),TRANS(1:6,1:6)
      INTEGER I
      DO 10  I=1,6
        TRANS(I,1)=XIDL(I,1)
10    CONTINUE
      DO 20  I=1,6
        TRANS(I,2)=XIDLS(I,1)
20    CONTINUE
      DO 30  I=1,6
        TRANS(I,3)=XIDG(I,1)
30    CONTINUE
      DO 40  I=1,6
        TRANS(I,4)=XIDGS(I,1)
40    CONTINUE
      DO 50  I=1,6
        TRANS(I,5)=XIDH(I,1)
50    CONTINUE
      DO 60  I=1,6
        TRANS(I,6)=XIDHS(I,1)
60    CONTINUE
      END


      POSVEL COMPUTES A POSITION AND A VELOCITY VECTOR IN THE PQW FRAME

      SUBROUTINE POSVEL(LSAT,GSAT,ECOSAT,ESAT,POS,VEL)
      REAL LSAT,GSAT,ECOSAT,ESAT,MU
      REAL POS(1:3,1:1),VEL(1:3,1:1)
      MU=1.
      POS(1,1)=((LSAT**2)/MU)*(COS(ESAT)-ECOSAT)
      POS(2,1)=(LSAT*GSAT/MU)*(SIN(ESAT))
      POS(3,1)=0
      VEL(1,1)=(-MU)*SIN(ESAT)/(LSAT*(1.-ECOSAT*COS(ESAT)))
```

```
      VEL(2,1)=(MU)*GS*T*SQRT(TRAT)/(LSAT **2*(1.-ECOS*TACOS(ESAT)))
      VEL(3,1)=0
      END
```

```
C     ROT33 COMPUTES A 3X3 ROTATION MATRIX FOR ROTATION FROM THE PQW
C     FRAME TO THE IJK FRAME

      SUBROUTINE ROT33(ROT)
      REAL ROT(1:3,1:3)
      REAL L,LS,G,GS,H,HS,E,ECC,MU
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
      ROT(1,1)=COS(HS)*COS(GS)-SIN(HS)*SIN(GS)*H/G
      ROT(1,2)=(-1. )*(COS(HS)*SIN(GS)+SIN(HS)*COS(GS)*H/G)
      ROT(1,3)=SIN(HS)*SQRT(1.0-H**2/G**2)
      ROT(2,1)=SIN(HS)*COS(GS)+COS(HS)*SIN(GS)*H/G
      ROT(2,2)=-SIN(HS)*SIN(GS)+COS(HS)*COS(GS)*H/G
      ROT(2,3)=-COS(HS)*SQRT(1.0-H**2/G**2)
      ROT(3,1)=SIN(GS)*SQRT(1.0-H**2/G**2)
      ROT(3,2)=COS(GS)*SQRT(1. -H**2/G**2)
      ROT(3,3)=H/G
      END
```

```
C     HM COMPUTES THE H MATRIX FROM RELATIVE POSITION AND VELOCITY  DATA

      SUBROUTINE HM(ST,HMAT)
      REAL RANGED,RRATED
      REAL ST(1:6,1:1),HMAT(1:2,1:6)
      RANGED=SQRT(ST(1,1)**2+ST(2,1)**2+ST(3,1)**2)
      RRATED=(ST(1,1)*ST(4,1)+ST(2,1)*ST(5,1)+ST(3,1)*ST(6,1))/RANGED
      HMAT(1,1)=ST(1,1)/RANGED
      HMAT(1,2)=ST(2,1)/RANGED
      HMAT(1,3)=ST(3,1)/RANGED
      HMAT(2,4)=HMAT(1,1)
      HMAT(2,5)=HMAT(1,2)
      HMAT(2,6)=HMAT(1,3)
      HMAT(1,4)=0
      HMAT(1,5)=0
      HMAT(1,6)=0
      HMAT(2,1)=(RANGED*ST(4,1)-RRATED*ST(1,1))/(RANGED**2)
      HMAT(2,2)=(RANGED*ST(5,1)-RRATED*ST(2,1))/(RANGED**2)
      HMAT(2,3)=(RANGED*ST(6,1)-RRATED*ST(3,1))/(RANGED**2)
      END
```

```
C     PHI COMPUTE THE STATE TRANSITION/COORDINATE TRANSFORMATION MATRIX
C     AT A GIVEN MEASUREMENTS TIME

      SUBROUTINE PHI(TRANS)
      REAL L,LS,G,GS,H,HS,E,ECC,MU,XPPL,XQPL,XWPL,VPPL,VQPL,VWPL,DEDL,
     *DEDLS,DXPDE,DXQDE,DXWDE,DVPDE,DVQDE,DVWDE,XPPG,XQPG,VPPG,VQP ,DEDG
      INTEGER I,J
      REAL R(1:6,1:6),PRPG(1:6,1:6),PRPGS(1:6,1:6),
     *PRPH(1:6,1:6),PRPHS(1:6,1:6),XPQW(1:6,1:1),DXDL(1:6,1:1),
     *DXDLS(1:6,1:1),DXDG(1:6,1:1),INTER(1:6,1:1),XIDL(1:6,1:1),
     *XIDLS(1:6,1:1),XIDG(1:6,1:1),XIDGS(1:6,1:1),XIDHS(1:6,1:1),
     *XIDH(1:6,1:1),TRANS(1:6,1:6)
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU

C     THIS SUBROUTINE COMPUTES  THE STATE TRANSITION DERIVATIVE MATRIX

      CALL ROTATE(R)
```

```
      CALL PART(PRPG,PRPGS,PRPH,PRPHS)
      CALL RVPOM(XPOM)
      CALL XDS(DXPDE,DXDDE,DXWDE,DVPDE,DVDDE,DVWDE)
      CALL XDL(DXPDE,DXDDE,DXWDE,DVPDE,DVDDE,DVWDE,DXDL)
      CALL XDLS(DXPDE,DXDDE,DXWDE,DVPDE,DVDDE,DVWDE,DXDLS)
      CALL XDG(DXPDE,DXDDE,DXWDE,DVPDE,DVDDE,DVWDE,DXDG)
      CALL IDL(P,DXDL,XIDL)
      CALL IDLS(P,DXDLS,XIDLS)
      CALL IDG(PRPG,XPOM,R,DXDG,XIDG)
      CALL IDGS(PRPGS,XPOM,XIDGS)
      CALL IDH(PRPH,XPOM,XIDH)
      CALL IDHS(PRPHS,XPOM,XIDHS)
      CALL FILL(XIDL,XIDLS,XIDG,XIDGS,XIDH,XIDHS,TRANS)
      END
      SUBROUTINE ADDN(NOSR,NOSRR,RA,RAR)
      SUBROUTINE ADDN ADDS NOISE TO MEASUREMENTS
      REAL NOSR,NOSRR,RA,RAR
        RA=RA+NOSR
        RAR=RAR+NOSRR
      END


      MATRIX-VECTOR MULTIPLICATION SUBROUTINE

      SUBROUTINE MULTI(A,B,PROD,N)
      INTEGER I,J,K,N
      REAL A(1:N,1:N),B(1:N,1:1),PROD(1:N,1:1)
      J=1
      DO 20 I=1,N
        PROD(I,J)=0
      DO 10 K=1,N
        PROD(I,J)=PROD(I,J)+A(I,K)*B(K,J)
10    CONTINUE
20    CONTINUE
      END


      MATRIX-MATRIX MULTIPLICATION SUBROUTINE

      SUBROUTINE MULT1(AA,BB,PROD1,M,N,OH)
      INTEGER I,J,K,M,N,OH
      REAL AA(1:N,1:M),BB(1:M,1:OH),PROD1(1:N,1:OH)
      DO 30 J=1,OH
      DO 20 I=1,N
        PROD1(I,J)=0.
      DO 10 K=1,M
        PROD1(I,J)=PROD1(I,J)+AA(I,K)*BB(K,J)
10    CONTINUE
20    CONTINUE
30    CONTINUE
      END


      SUBROUTINE KEPLER USES THE KEPLER ALGORITHM TO FIND THE
      ECCENTRIC ANOMALY OF AN ORBIT GIVEN A PARTICULAR TIME PAST
      PERIGEE OF THE ORBIT.
      SUBROUTINE KEPLER(TIM,ECC1,ECENT)
      REAL L,LS,G,GS,H,HS,E,ECC,MU,TIM,ECENT,M,M1,DMDE,TEST
      COMMON /ELEM/ L,LS,G,GS,H,HS,E,ECC,MU
      M=(MU**2/L**3)*TIM
      ECENT=M
10    M1=ECENT-ECC1*SIN(ECENT)
      DMDE=1.-ECC1*COS(ECENT)
      ECENT=ECENT+(M-M1)/DMDE
```

```
      TEST=M-M1
      TEST=ABS(TEST)
      IF (TEST .LT. 1E-5 ) GO TO 20
      GO TO 1
20    CONTINUE
      END


      MPRINT PRINTS A MATRIX OR VECTOR

      SUBROUTINE MPRINT(A,RR,C,RMAX)
      INTEGER RR,C,I,J,RMAX
      REAL A(RMAX,1)
      DO 10 I=1,RR
        PRINT '(3X,6E11.3)',(A(I,J),J=1,C)
10    CONTINUE
      PRINT *,' '
      PRINT *,' '
      END


      TRPOS COMPUTES THE TRANSPOSE OF A MATRIX

      SUBROUTINE TRPOS(MAT,MATT,ROW,COL,RMAX,CMAX)
      INTEGER ROW,COL,I,J,RMAX,CMAX
      REAL MAT(RMAX,1),MATT(CMAX,1)
      DO 1  I=1,ROW
      DO 1  J=1,COL
        MATT(J,I)=MAT(I,J)
10    CONTINUE
      END


      MATRIX-MATRIX MULTIPLICATION SUBROUTINE

      SUBROUTINE MULT1A(AA,BB,PROD1,M,N,OH,NMAX,MMAX)
      INTEGER I,J,K,M,N,OH,NMAX,MMAX
      REAL AA(NMAX,1),BB(MMAX,1),PROD1(NMAX,1)
      DO 3  J=1,OH
      DO 2  I=1,N
        PROD1(I,J)=0.
      DO 1  K=1,M
        PROD1(I,J)=PROD1(I,J)+AA(I,K)*BB(K,J)
10    CONTINUE
20    CONTINUE
30    CONTINUE
      END

      SUBROUTINE MULTIA(A,B,PROD,N,NMAX)
      INTEGER I,J,K,N,NMAX
      REAL A(NMAX,1),B(NMAX,1),PROD(NMAX,1)
      J=1
      DO 2  I=1,N
        PROD(I,J)=0.
      DO 1  K=1,N
        PROD(I,J)=PROD(I,J)+A(I,K)*B(K,J)
10    CONTINUE
20    CONTINUE
      END
```

VITA

John F. Anthony was born on 24 April 1956 in Paterson, New Jersey. He graduated from high school in Hawthorne, New Jersey in 1974 and attended the U.S. Air Force Academy, Colorado, from which he received a Bachelor of Science degree in Astronautical Engineering in May 1978. He was assigned to the Flight Test Section of the 6585 Test Group at Holloman Air Force Base, New Mexico, where he served as a flight test engineer on NC-141, NC-130, RF4-C, and UH-1H aircraft. While in New Mexico he also served as a search pilot for the New Mexico Civil Air Patrol. He entered the Air Force Institute of Technology in June 1980.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/GA/AA/81D-1	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Design of an Orbital Element Estimator Using Relative Motion Data	5. TYPE OF REPORT & PERIOD COVERED: MS Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): John F. Anthony, 1Lt, USAF	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT-EN) Wright-Patterson Air Force Base, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: December 1981
	13. NUMBER OF PAGES: 103
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW 190-17

FREDRIC C. LYNCH, Major,USAF
Director of Public Affairs

Fredric C Lynch

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Least Squares Estimation
Orbital Elements
Bayes Estimation
Relative Motion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A relative orbital element estimator is designed using least squares estimation. Range and range rate measurements are taken and a vector of Delaunay elements, relative to an interceptor satellite, is estimated using the nonlinear least squares equation. The estimator incorporates a state vector coordinate transformation from relative position and velocity to relative orbital elements. Two modes of the estimator program are designed, batch and sequential, where the sequential mode uses Bayes estimation. Three test cases are analyzed and indicate satisfactory performance.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Problem areas include estimator dependence upon a highly accurate initial estimate of the element vector to start the estimation process. The orbits are restricted to noncircular forboth satellites and the orbits must be non coplanar. Range and range rate for the estimator are provided using a truth model.

FILMED

3-8